

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO**

**SISTEMA DE INFORMAÇÃO PARA APURAÇÃO DE PONTO  
ELETRÔNICO UTILIZANDO BANCO DE DADOS POSTGRESQL**

**MARCELO JOSÉ GONÇALVES**

**BLUMENAU**  
**2004**

**2004/2-34**

**MARCELO JOSÉ GONÇALVES**

**SISTEMA DE INFORMAÇÃO PARA APURAÇÃO DE PONTO  
ELETRÔNICO UTILIZANDO BANCO DE DADOS  
POSTGRESQL**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Ciência  
da Computação — Bacharelado.

Prof. Alexander Roberto Valdameri - Orientador

**BLUMENAU  
2004**

**2004/2-34**

**SISTEMA DE INFORMAÇÃO PARA APURAÇÃO DE PONTO  
ELETRÔNICO UTILIZANDO BANCO DE DADOS  
POSTGRESQL**

Por

**MARCELO JOSÉ GONÇALVES**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada  
por:

Presidente:

---

Prof. Alexander Roberto Valdameri – Orientador, FURB

Membro:

---

Prof. Wilson Pedro Carli, FURB

Membro:

---

Prof. Everaldo Artur Grahl, FURB

Blumenau, 07 de Dezembro de 2004

Dedico este trabalho a toda minha família e especialmente a minha noiva por terem me compreendido nos momentos de dificuldade da realização deste.

## AGRADECIMENTOS

À Deus, por me conceder a vida.

À minha família, em especial meu pai José e minha mãe Nilsa por terem me acolhido durante este semestre.

À minha noiva Gilmara, por ter me compreendido nos momentos que estava estressado.

Ao amigo Gilmar Cardoso, pelo tempo em que passamos juntos definindo as regras para a apuração do ponto.

À empresa para o qual o trabalho foi desenvolvido, que me disponibilizou as informações necessárias para implementação do sistema.

Ao amigo Luiz Augusto Spranger, por me passar um pouco do seu vasto conhecimento na computação.

Ao meu orientador, Prof. Alexander Roberto Valdameri, por ter acreditado na relevância deste trabalho e por ter me apoiado para conclusão deste.

## RESUMO

Este trabalho apresenta a implementação de um sistema de informação para apuração de ponto eletrônico, mais especificamente para uma empresa de construção civil. Descreve o desenvolvimento do sistema utilizando a biblioteca ZeosDBO, o sistema gerenciador de banco de dados PostgreSQL e o emulador do sistema operacional Cygwin como soluções *open source*. O projeto do sistema de informação aplica a metodologia da análise essencial.

Palavras chaves: Software livre; PostgreSQL; Banco de dados; Ponto eletrônico.

## **ABSTRACT**

This work presents the implementation of a system of information for verification of electronic point, more specifically for a company of civil construction. It describes the development of the system using the ZeosDBO library, the database management system PostgreSQL and the emulator of the operational system Cygwin as solutions open source. The project of the information system applies the methodology of the essential analysis.

Key-Words: Free Software; PostgreSQL; Database; Electronic Point.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de contexto do SI.....	24
Figura 2 – DFD – Usuário mantém Cidade.....	25
Figura 3 – DFD – Usuário mantém Empresa.....	25
Figura 4 – DFD – Usuário mantém Situação.....	25
Figura 5 – DFD – Usuário mantém Cálculo.....	25
Figura 6 – DFD – Usuário mantém Feriado.....	26
Figura 7 – DFD – Usuário mantém Datas Feriado.....	26
Figura 8 – DFD – Usuário mantém Escala.....	26
Figura 9 – DFD – Usuário mantém Datas Escala.....	27
Figura 10 – DFD – Usuário mantém Relógio.....	27
Figura 11 – DFD – Usuário mantém Local.....	27
Figura 12 – DFD – Usuário mantém Afastamento.....	28
Figura 13 – DFD – Usuário mantém Marcação Original.....	28
Figura 14 – DFD – Usuário mantém Marcação.....	28
Figura 15 – DFD – Usuário mantém Ocorrências Marcação.....	29
Figura 16 – DFD – Usuário mantém Situação Apurada.....	29
Figura 17 – DFD – Usuário mantém Banco de Horas.....	30
Figura 18 – DFD – Usuário mantém Sub-empresa.....	30
Figura 19 – DFD – Usuário mantém Colaborador.....	30
Figura 20 – DFD – Usuário mantém Função.....	31
Figura 21 – DFD – Usuário emite Rel. de Situações Apuradas.....	31
Figura 22 – DFD – Usuário emite Rel. Espelho de Cartão Ponto.....	32
Figura 23 – DFD – Usuário emite Relatório de Horas para Faturamento.....	32
Figura 24 – DFD – usuário executa rotina de apuração/cálculo de horas.....	33
Figura 25 – DER – Modelo Conceitual.....	34
Figura 26 – DER – Modelo Físico.....	35
Figura 27 – Paleta de componentes do ZeosDBO.....	37
Quadro 1 – Método pEfetuaConexao da classe TfrLogin.....	37
Quadro 2 – Trecho da <i>function</i> fApuraSituacao.....	39
Quadro 3 – Fluxograma - Algoritmo de Apuração.....	40
Quadro 4 – Diagrama de Interação do Sistema.....	41
Figura 28 – Interface Selecciona Empresa/Período de Apuração.....	41
Figura 29 – Barra de <i>Status</i> do Sistema.....	41
Figura 30 – Interface Cidades.....	42
Figura 31 – Interface Escalas de Horário.....	43
Figura 32 – Interface Locais.....	45
Figura 33 – Interface Situações.....	46
Figura 34 – Interface Colaboradores.....	48
Figura 35 – Interface Apurar Ponto.....	49
Figura 36 – Relatório de Situações Apuradas.....	50
Figura 37 – Interface Registro de Ponto.....	50
Figura 38 – Interface Afastamentos.....	52

## LISTA DE TABELAS

Tabela 1 – Características dos SPT.....	16
Tabela 2 – Registro de Ponto.....	53
Tabela 3 – Situações Apuradas e Rateadas.....	54

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 OBJETIVOS DO TRABALHO .....	11
1.2 MOTIVAÇÃO.....	12
1.3 ESTRUTURA DO TRABALHO .....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1 SISTEMAS DE INFORMAÇÃO.....	14
2.2 SISTEMAS DE PROCESSAMENTO DE TRANSAÇÕES .....	15
2.3 SOFTWARE LIVRE.....	16
2.4 BANCO DE DADOS E SGBD .....	18
2.5 BANCO DE DADOS POSTGRESQL.....	18
2.5.1 Histórico.....	19
2.5.2 Características .....	19
2.5.3 Vantagens.....	20
2.6 SISTEMA DE APURAÇÃO DE PONTO ELETRÔNICO.....	20
2.7 HORÁRIO DE TRABALHO, HORAS NORMAIS, EXTRAS E FALTAS.....	21
<b>3 DESENVOLVIMENTO DO TRABALHO .....</b>	<b>22</b>
3.1 REQUISITOS .....	22
3.2 ESPECIFICAÇÃO .....	22
3.2.1 Lista de Eventos .....	23
3.2.2 Diagrama de Contexto.....	23
3.2.3 Diagrama de Fluxo de Dados.....	24
3.2.4 Diagrama Entidade-Relacionamento – Modelo Conceitual.....	33
3.2.5 Diagrama Entidade-Relacionamento – Modelo Físico .....	35
3.3 IMPLEMENTAÇÃO .....	36
3.3.1 Técnicas e ferramentas utilizadas.....	36
3.3.2 Operacionalidade da implementação .....	40
3.4 RESULTADOS E DISCUSSÃO .....	53
<b>4 CONCLUSÕES .....</b>	<b>55</b>
4.1 EXTENSÕES .....	55
APÊNDICE A – Dicionário de Dados. ....	59
APÊNDICE B – Passos para instalação do Cygwin no Windows. ....	65
APÊNDICE C – Método pInsertCommit da classe TfrEscolhe.....	66

APÊNDICE D – Trecho do método pFillFields da classe TfrEscolhe.....	68
APÊNDICE E – Relação das situações tratadas na apuração. ....	69

## 1 INTRODUÇÃO

Com a rápida evolução das tecnologias na computação, fica difícil nos dias atuais para as pequenas e médias empresas renovarem seu parque de Tecnologia da Informação (TI) constantemente. Dentro deste contexto, o emprego de software livre (de livre distribuição) dentro das empresas e órgãos governamentais tem sido uma alternativa para essas instituições se manterem atualizadas e para diminuir os custos com investimento em software.

Com a expansão da Internet, disseminaram-se várias comunidades com o intuito de implementar e difundir o software livre. Na Internet estão disponibilizados vários softwares livres que podem ser copiados e usados sem a necessidade de se pagar por isso. Existem softwares livres para vários fins, tais como, pacotes para automação de escritório, antivírus, softwares educacionais, softwares de entretenimento, sistemas operacionais, banco de dados e outros.

Dentre os Sistemas Gerenciadores de Banco de Dados (SGBD), o PostgreSQL tem-se destacado. De acordo com Momjian (2001, p. 1, tradução nossa), PostgreSQL é o mais avançado servidor de banco de dados *open source* (código fonte aberto). Conforme Neves (2002, p. 18), “é um dos melhores bancos de dados para sistema operacional GNU/Linux, voltado para aplicações simples ou complexas com uma robustez excepcional [...]”

Acreditando nisto e visando a implantação de solução viável e de baixo custo, este trabalho apresenta um Sistema de Informação (SI) utilizando o SGBD PostgreSQL, cujo objetivo é automatizar o controle dos colaboradores de uma empresa do setor de construção civil, sediada em Blumenau/SC, fundada em setembro de 1942 e que conta atualmente com aproximadamente 800 colaboradores em seu quadro. A empresa presta serviços principalmente para a indústria, construindo galpões, edifícios comerciais e administrativos, estações de tratamento de efluentes, laboratórios, pontes e outros.

Devido ao grande número de colaboradores, à alta rotatividade destes e por seus canteiros de obra serem dispersos geograficamente, a administração de pessoal é uma tarefa difícil de ser executada, principalmente no que se refere ao controle de frequência de ponto.

O controle de frequência de ponto atualmente é feito de maneira convencional. Mensalmente são distribuídos os cartões ponto nas obras para que diariamente sejam registradas a entrada e a saída. Diariamente pode-se ter vários registros de entrada e/ou saída

no cartão ponto, uma vez que os colaboradores ausentam-se de seu local de trabalho e posteriormente retornam ou quando da transferência deles para outro local. A transferência de pessoal de um local de trabalho para outro geralmente é feita diretamente pelos encarregados das obras, ou seja, o colaborador sai de um local e vai para outro sem que o departamento pessoal da empresa tenha conhecimento. Muitas vezes pergunta-se “Onde o colaborador está no momento?”, “Em qual escala de horário ele está trabalhando?”. Essas informações não são precisas atualmente, porque o departamento não tem a informação diária destes acontecimentos. Para tentar resolver essas situações, a empresa vem a alguns anos tentando implantar uma solução para o controle de frequência de ponto.

Em 1991, a construtora adquiriu, de uma Software House de São Paulo/SP, um software de apuração de ponto eletrônico, o qual a mesma se propôs a adaptá-lo. Em 1993, o investimento foi abandonado, porque o software foi tão adaptado que ficou praticamente impossível um usuário comum trabalhar com o mesmo. Desde 1997 a empresa vem tentando implantar um conceituado software de apuração de ponto eletrônico de uma Software House de Blumenau/SC, mas pelo fato do sistema ter que ser modificado para atender as necessidades específicas descritas anteriormente, decidiu-se desenvolver um novo sistema para implantar uma solução de baixo custo para resolver seus problemas operacionais.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um software de apuração de ponto eletrônico para uma empresa do ramo de construção civil da região, utilizando o banco de dados PostgreSQL como repositório para os dados.

Os objetivos específicos do trabalho são:

- a) desenvolver um algoritmo para o tratamento das horas dos colaboradores conforme horário e local de trabalho, para calcular a quantidade de horas trabalhadas, extras, faltas, saídas antecipadas, atrasos, compensadas, feriados, afastamentos;
- b) efetuar o rateio das horas calculadas por local de trabalho;
- c) efetuar automaticamente a troca de escala de horário conforme o local de trabalho do registro de ponto dos colaboradores.

## 1.2 MOTIVAÇÃO

O desenvolvimento deste trabalho justifica-se pelo fato da empresa não ter encontrado um software de apuração de ponto eletrônico que resolva as situações específicas existentes no processo de apontamento de horas dos colaboradores, que difere das empresas de outros ramos de atividade, pela grande rotatividade de pessoal em seus canteiros de obra. Isto é, não existe solução computacional no mercado de software.

O trabalho irá contribuir para a automatização do processo de apontamento manual de cartão ponto e o controle de pessoal nas obras, permitindo:

- a) eliminação de erros no cálculo manual quanto ao tratamento de horas, tais como horas trabalhadas, extras, faltas, saídas antecipadas, atrasos, compensadas, feriados, afastamentos;
- b) eliminação da transcrição das horas apuradas para formulário em papel;
- c) eliminação da digitação das horas apuradas no sistema de folha de pagamento;
- d) rateio das horas por local de trabalho, para prover informações mais detalhadas para folha de pagamento e faturamento de mão de obra;
- e) automatização da troca de horário de trabalho conforme o local do registro de ponto;
- f) redução no tempo atual de dois dias para apenas alguns minutos no processo de apuração de ponto.

## 1.3 ESTRUTURA DO TRABALHO

O trabalho está organizado e dividido em quatro capítulos.

O primeiro capítulo faz uma introdução sobre os assuntos abordados descrevendo os objetivos e a motivação para o desenvolvimento.

No segundo capítulo são fundamentados os assuntos relevantes a realização do trabalho: sistema de informação (SI), sistema de processamento de transações (SPT), software livre, banco de dados e sistema gerenciador de banco de dados (SGBD), sistema de apuração de ponto eletrônico, PostgreSQL e conceitos de horas.

O terceiro capítulo descreve os requisitos, a especificação, a implementação, a operacionalidade do sistema desenvolvido e os resultados e discussão.

No quarto e último capítulo são apresentadas às conclusões do trabalho e sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

A revisão bibliográfica vem abordar os assuntos relevantes ao desenvolvimento deste trabalho, portanto na primeira seção são citados conceitos de SI. A segunda seção apresenta conceitos sobre sistemas de processamento de transações (SPT). Na terceira seção são contextualizados a origem, conceitos e definições de algumas organizações em relação ao software livre. Na quarta seção são citados conceitos de banco de dados e sistemas gerenciadores de banco de dados, para que na quinta seção seja apresentado o banco de dados PostgreSQL, que é um software livre. São apresentados o histórico, características e vantagens. A sexta seção descreve algumas características de softwares de apuração de ponto eletrônico consultados. Na sétima e última seção são apresentados conceitos para o tratamento de horas.

### 2.1 SISTEMAS DE INFORMAÇÃO

A informática ou TI pode ser conceituada como recursos tecnológicos e computacionais para guarda, geração e uso da informação e do conhecimento. Todo sistema, usando ou não os recursos da TI, que manipula e gera informação pode ser genericamente considerado SI (REZENDE, 2003, p. 68).

Conforme Stair e Reynolds (2002, p. 12) SI é um tipo especial de sistema composto por um conjunto de elementos ou componentes inter-relacionados que coletam (entrada), manipulam (processamento) e disseminam (saída) os dados e a informação e fornecem um mecanismo de *feedback* para atender um objetivo.

Laudon K. e Laudon J. (2004, p. 39 - 40) classificam os SI em quatro níveis principais:

- a) sistemas do nível operacional: dão suporte aos gerentes operacionais, acompanhando atividades e transações elementares da organização. Ex: sistema que acompanhe o número de horas trabalhadas por dia pelos trabalhadores do chão de fábrica;
- b) sistemas do nível do conhecimento: dão suporte aos trabalhadores do conhecimento e de dados da organização. Seu propósito é auxiliar a empresa comercial a integrar novas tecnologias ao negócio e ajudar a organização a controlar o fluxo de documentos;

- c) sistemas de nível gerencial: atendem às atividades de monitoração, controle, tomada de decisões e procedimentos administrativos dos gerentes médios;
- d) sistemas do nível estratégico: auxiliam a gerência sênior a atacar e enfrentar questões estratégicas e tendências de longo prazo tanto na empresa quanto no ambiente externo.

Conforme Laudon K. e Laudon J. (2004, p. 41 - 42) os seis tipos mais importantes de SI são:

- a) Sistemas de Apoio ao Executivo (SAE) - no nível estratégico;
- b) Sistemas de Informações Gerenciais (SIG) - no nível estratégico;
- c) Sistema de Apoio a Decisão (SAD) – no nível gerencial;
- d) Sistemas de Trabalhadores do Conhecimento (STC) – no nível gerencial;
- e) Sistemas de Automação de Escritório – no nível do conhecimento;
- f) Sistemas de Processamento de Transações (SPT) – no nível operacional.

A seguir será fundamentado SPT que é o tipo do SI implementado neste trabalho. Maiores informações referente aos outros tipos citados podem ser obtidos na obra de LAUDON K. e LAUDON J. (2004, p. 41 - 71)

## 2.2 SISTEMAS DE PROCESSAMENTO DE TRANSAÇÕES

Um Sistema de Processamento de Transações caracteriza-se por registrar as transações rotineiras necessárias ao funcionamento da empresa. São em geral, tão críticos que, se deixarem de funcionar por algumas horas, podem causar dano a empresa. Os gerentes necessitam de SPT para monitorar o andamento das operações internas e as relações externas da empresa com o ambiente externo. Os SPT produzem informações para outros tipos de sistemas (LAUDON K.; LAUDON J., 2004, p. 41 - 42).

Um SPT é o mais baixo nível de sistemas de informações, atendendo às necessidades do nível operacional da organização. É utilizado pelos profissionais da empresa em todos os níveis de execução. Esse sistema tem como função executar e cumprir os planos elaborados por todos os outros sistemas, pois serve como base na entrada de dados (ROSINI; PALMISANO, 2003, p. 15).

Os SPT contemplam o processo de operações e transações rotineiras cotidianas, incluindo seus respectivos procedimentos. Manipulam informações em seu detalhe, analítica,

no singular. Por exemplo: nome do produto, tipo do produto, data da venda (REZENDE, 2003, p. 62.).

As principais características dos SPT demonstradas na Tabela 1 são (ROSINI; PALMISANO, 2003, p. 15 – 16; LAUDON K.; LAUDON J., 2004, p. 42):

- a) identificação do evento (transação);
- b) têm como finalidade: intercalar, listar, ordenar, atualizar;
- c) possibilidade de criar relatórios detalhados, lista e sumário;
- d) pode ser utilizado em todos os níveis de execução da empresa por atender às necessidades do nível operacional da organização.

Tabela 1 – Características dos SPT

ENTRADA	PROCESSAMENTO	SAÍDA	USUÁRIOS
Evento, transação	Intercalar, listar, ordenar, atualizar	Relatório detalhado, lista, sumário	Operário, nível básico da estrutura, seus supervisores

### 2.3 SOFTWARE LIVRE

Existem várias interpretações para o termo software livre que, conforme Momjian (2001, p. 4, tradução nossa), frequentemente confundem as pessoas. Com a comunicação via Internet, várias organizações mantêm *sites* que servem de ponte para desenvolvedores do mundo inteiro trocarem conhecimento e difundirem o software livre.

Conforme a FSF (2002a), um dos precursores do software livre foi Richard Stallman que em 1984 escreveu “O Manifesto GNU”, que trata do projeto *GNU's Not Unix* (GNU). O projeto GNU, que significa Gnu Não é Unix, mantido atualmente pela Free Software Foundation (FSF), descreve a intenção inicial de Stallman de compartilhar seu conhecimento com as outras pessoas e de desenvolver um sistema operacional completo compatível com o Unix. GNU é um acrônimo recursivo que se pronuncia "guh-NEW" ou "guniw".

As licenças de muitos softwares são desenvolvidas para restringir sua liberdade de compartilhá-los e mudá-los. Contrária a isso, a licença GNU *General Public Licence* (GPL), pretende garantir a liberdade de compartilhar e alterar softwares livres, garantindo que o software será livre e gratuito para os usuários (FSF, 2004a, tradução nossa). Para a distribuição e utilização de bibliotecas, pode-se utilizar a licença GNU *Lesser General Public*

*License* (LGPL), anteriormente conhecida como GNU *Library General Public Licence*, que ao contrário da GNU GPL, permite a utilização de biblioteca em programas proprietários. O uso da licença GNU GPL em bibliotecas torna a mesma livre somente para software livre (FSF, 2004b, tradução nossa).

De acordo com a FSF (2002b), a palavra livre está relacionada com liberdade e não com o preço. Portanto, pode-se ou não pagar um preço para adquirir um software livre.

Software livre se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro tipos de liberdade, para os usuários do software:

- liberdade de executar o programa, para qualquer propósito;
- liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades;
- liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo;
- liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie.

Um programa é software livre se os usuários têm todas essas liberdades. Portanto, o usuário deve ser livre para redistribuir cópias, seja com ou sem modificações, seja de graça ou cobrando uma taxa pela distribuição, para qualquer um em qualquer lugar. Ser livre para fazer essas coisas significa (entre outras coisas) que você não tem que pedir ou pagar pela permissão. [...]

Software livre não significa "não-comercial". Um programa livre deve estar disponível para uso, desenvolvimento, e distribuição comercial. O desenvolvimento comercial de software livre não é incomum; tais softwares livres comerciais são muito importantes. (FSF, 2003).

Para OSI (2004a, tradução nossa), quando os programadores podem ler, redistribuir ou modificar o código fonte, o software evolui. As pessoas melhoram-no, adaptam-no e corrigem *bugs*. Isto pode acontecer a uma velocidade que, se comparado ao desenvolvimento convencional de software, parece impossível. A Open Source Initiative (OSI), organização sem fins lucrativos que certifica software livre, entende que este processo de desenvolvimento produz softwares melhores e mais rápido que o modelo fechado tradicional, onde poucos têm acesso ao código fonte. A OSI trabalha para que isto seja possível comercialmente.

Conforme OSI (2004b, tradução nossa), as licenças “clássicas” mais utilizadas para software livre, antes da liberação da Mozilla em meados de 1998, são GPL, LGPL, *Berkeley Software Distribution* (BSD) e Massachusetts Institute of Technology (MIT). A licença pública Mozilla vem desde então sendo largamente utilizada.

## 2.4 BANCO DE DADOS E SGBD

De acordo com Date (1991, p. 5), “o sistema de banco de dados é basicamente um sistema de manutenção de registros por computador – ou seja, um sistema cujo objetivo global é manter as informações e torná-las disponíveis quando solicitadas.”

Para Neves (2002, p. 17), “o SGBD permite que bancos de dados “persistentes” sejam concorrentemente partilhados por usuários e aplicações utilizando manuseio de armazenamento e estratégias de otimização”.

Dentre as abordagens de banco de dados a mais difundida é a abordagem relacional. Um dos bancos de dados objeto-relacional *open source* que mais tem evoluído e popularizado é o PostgreSQL.

## 2.5 BANCO DE DADOS POSTGRESQL

O SGBD PostgreSQL é distribuído sob a licença BSD, reconhecida pela OSI (POSTGRESQL, 2004, tradução nossa). O PostgreSQL é o mais avançado servidor de banco de dados *open source* (MOMJIAN, 2001, p. 1, tradução nossa). Conforme Neves (2002, p. 18), é um SGBD relacional e orientado a objetos.

PostgreSQL é um sofisticado Sistema Gerenciador de Banco de Dados Objeto-Relacional (SGBDOR). Um SGBDOR é uma extensão do tradicional Sistema Gerenciador de Banco de Dados Relacional (SGBDR). O aspecto objeto-relacional do PostgreSQL adiciona inúmeras funcionalidades diretamente ao modelo de dados relacional. Isto inclui suporte para *arrays* (múltiplos valores em uma única coluna), herança (relacionamento pai-filha entre tabela) (WORSLEY & DRAKE, 2002, p. 35 - 36, tradução nossa).

O PostgreSQL pode ser instalado em vários sistemas operacionais (SO), tais como: Windows 95/98/NT/ME/2000/XP, Mac OS X, FreeBSD, Unix (34 plataformas na última versão) e outros baseados em GNU/Linux (DBEXPERTS, 2004; ADVOCACY, 2002).

O suporte nativo ao SO Windows está em fase de teste, portanto, para executar o PostgreSQL neste ambiente é necessário a instalação do software Cygwin. Conforme Cygwin (2004b, tradução nossa) o Cygwin é um software livre distribuído sob licença GNU GPL. Cygwin é um software capaz de emular o SO Unix no ambiente Windows

95/98/NT/ME/2000/XP. O emulador disponibiliza através da dll "cygwin1.dll" as principais funcionalidades do ambiente Unix (CYGWIN, 2004a, tradução nossa).

### 2.5.1 Histórico

O PostgreSQL é resultado de um trabalho que iniciou com o Ingres, que foi desenvolvido na Universidade da Califórnia Berkeley entre 1987 e 1995. O Ingres foi posteriormente aperfeiçoado pela Relational Technologies/Ingres Corporation, que produziu um dos primeiros servidores de banco de dados de sucesso comercial. Ainda na Berkeley, Michael Somebraker coordenou uma equipe para desenvolver um servidor de banco de dados objeto-relacional chamado Postgres. Dois estudantes graduados da Berkeley, Jolly Chen e Andrew Yu, posteriormente adicionaram um interpretador SQL (*Structured Query Language*) ao Postgres. Em 1995, foi lançada a versão Postgre95 e, em 1996, a versão PostgreSQL (MOMJIAN, 2001, p. 1, tradução nossa). Atualmente o seu desenvolvimento é mantido e coordenado pelo Grupo Global de Desenvolvimento do PostgreSQL, na WEB, que conta com a cooperação de uma comunidade formada por empresas e pessoas espalhadas por vários continentes do mundo.

### 2.5.2 Características

A seguir são apresentadas algumas características técnicas que o PostgreSQL possibilita (DBEXPERTS, 2004; ADVOCACY, 2002):

- a) sintaxe compatível com o padrão ANSI SQL 89, 92 e 99;
- b) gerenciamento de transações totalmente consistente com o padrão ACID (Atomicidade, Consistência, Isolamento e Durabilidade);
- c) herança;
- d) funções (*store procedures*) podem ser escritas pelo usuário com interface nativa em várias linguagens, tais como C, SQL, PL/pgSQL, TCL, Perl, Python e Ruby;
- e) suporte nativo a *Secure Socket Layer* (SSL);
- f) sistema de tipagem de dados expansível, provendo customização e criação de novos tipos de dados definidos pelo usuário;
- g) extensa biblioteca de funções e operadores, com centenas de funções pré-implementadas (matemáticas, data/hora, *string*, geométricas, formatação etc.);

- h) sistema de controle de concorrência de versão múltipla (MVCC) para aplicações concorrentes altamente escaláveis, onde a leitura não bloqueia a escrita e vice-versa;
- i) tamanho virtualmente ilimitado do banco de dados e tabelas, isto é, número ilimitado de registros e índices por tabela.

### 2.5.3 Vantagens

Conforme Dbexperts (2004) e a Advocacy (2002), o PostgreSQL oferece muitas vantagens, onde entre estas cita-se:

- a) não existem custos para licenciamento do software;
- b) o código fonte está disponível, permitindo a customização se for necessário;
- c) diferente de muitos bancos de dados proprietários é comum relatos de entidades afirmando que o PostgreSQL nunca parou de funcionar em muitos anos de operação com muita atividade;
- d) utiliza a estratégia de armazenamento MVCC que o torna extremamente estável em ambientes com grande volume de dados.

## 2.6 SISTEMA DE APURAÇÃO DE PONTO ELETRÔNICO

Existem vários sistemas de apuração de ponto eletrônico para o apontamento das ocorrências diárias no que se refere ao registro entrada e/ou saída de cartão ponto. Os sistemas consultados possibilitam controle das horas diárias computadas, tais como: horas trabalhadas, extras, faltas, saída antecipada, atrasos, compensadas, feriados, afastamentos etc.

Os sistemas Tecnoponto (DIGILOGO, 2004), Newponto (NORBER, 2004), Sipe (DEFAULT, 2004), Winpto (LCS, 2004) dispõem de um módulo para controle do banco de horas dos colaboradores. O sistema Forponto (TASK, 2004) possibilita o acompanhamento dos custos com a mão-de-obra. Os sistemas oferecem integração com sistema de folha de pagamento, ou seja, geram a quantidade de horas computadas para serem posteriormente importadas e calculados os holerites dos mesmos. Todos disponibilizam relatórios para conferência das ocorrências diárias do cartão ponto e opções para que sejam abonadas ou justificadas as horas faltantes ou excedentes.

Em relação ao repositório dos dados, o sistema Forponto pode ser utilizado com o banco de dados Oracle, SQL Server, DB2 e Sybase. O sistema Sipe pode ser utilizado com o

banco de dados Oracle, SQL Server e Access. Os outros sistemas consultados não descrevem o banco de dados que utilizam.

## 2.7 HORÁRIO DE TRABALHO, HORAS NORMAIS, EXTRAS E FALTAS

Conforme Nascimento (2004, p. 323) o horário de trabalho abrange a indicação da hora que se inicia até a hora em que termina o trabalho. O estudo da jornada diária de trabalho compreende não só a duração do trabalho, mas os horários, o intervalo e outros aspectos [...].

Como regra geral, a duração do trabalho normal não pode exceder a 8 horas diárias e 44 semanais, facultada a compensação de horários e a redução da jornada mediante acordo ou convenção coletiva do trabalho (NASCIMENTO, 2004, p. 324).

Horas extras são aquelas que excedem a jornada normal fixada por lei, convenção coletiva, sentença normativa ou contrato individual de trabalho. Justificadas são as faltas que o empregado, por lei, convenção coletiva do trabalho, regulamento de empresa ou contrato individual de trabalho, pode dar, sem prejuízo da sua remuneração (NASCIMENTO, 2004, p. 313, 357).

### 3 DESENVOLVIMENTO DO TRABALHO

De acordo com os assuntos abordados na introdução e na fundamentação teórica, na primeira seção deste capítulo são descritos os requisitos do sistema, na segunda seção é descrita a especificação com lista de eventos e diagramas, na terceira seção é demonstrada a implementação descrevendo técnicas, ferramentas utilizadas e a operacionalidade do sistema. Na última seção são apresentados resultados e discussões.

#### 3.1 REQUISITOS

A elicitação dos requisitos foi feita em conjunto com o departamento de recursos humanos da construtora, visitando obras e acompanhando o processo de apontamento de cartões ponto manual atualmente utilizado.

O sistema deverá:

- a) importar arquivo formato texto (ASCII) gerado por relógio-ponto e/ou coletor de dados dos registros de ponto dos colaboradores (requisito funcional - RF);
- b) importar arquivo formato texto (ASCII) gerado pelo sistema de folha de pagamento existente com registros pertinentes aos colaboradores (RF);
- c) calcular a quantidade de horas diárias trabalhadas, extras, faltas, feriados, descanso semanal remunerado (RF);
- d) efetuar o rateio das horas por local de trabalho (RF);
- e) efetuar a troca de horário de trabalho conforme local do registro de ponto eletrônico (RF);
- f) gerar arquivo texto com eventos rateados por local de trabalho para ser importado no sistema de folha de pagamento (RF);
- g) gerar advertências quando da falta de marcação no registro de ponto (RF);
- h) utilizar a linguagem de programação *Object Pascal* (requisito não funcional - RNF);
- i) utilizar o banco de dados PostgreSQL (RNF).

#### 3.2 ESPECIFICAÇÃO

O sistema foi projetado utilizando a ferramenta Power Designer 10, seguindo a metodologia de análise essencial, onde são definidos as listas de eventos, os diagramas de contexto, diagramas de fluxo de dados e o modelo entidade-relacionamento (MER).

### 3.2.1 Lista de Eventos

A seguir são apresentados os eventos que compõe o sistema desenvolvido:

- a) usuário mantém Cidade;
- b) usuário mantém Empresa;
- c) usuário mantém Situação;
- d) usuário mantém Cálculo;
- e) usuário mantém Feriado;
- f) usuário mantém Datas Feriado;
- g) usuário mantém Escala;
- h) usuário mantém Datas Escala;
- i) usuário mantém Relógio/Coletor;
- j) usuário mantém Local;
- k) usuário mantém Afastamento;
- l) usuário mantém Marcações Originais;
- m) usuário mantém Marcação;
- n) usuário mantém Ocorrências Marcação;
- o) usuário mantém Situação Apurada;
- p) usuário mantém Banco de Horas;
- q) usuário importa Sub-Empresa;
- r) usuário importa Função;
- s) usuário importa Colaborador;
- t) usuário emite relatório das situações apuradas;
- u) usuário emite relatório espelho do cartão ponto;
- v) usuário emite relatório de horas para faturamento;
- w) usuário executa rotina de apuração/cálculo de horas.

### 3.2.2 Diagrama de Contexto

O objetivo do SI de apuração de ponto eletrônico aqui implementado é fazer o apontamento/cálculo de horas diárias dos colaboradores fazendo o rateio por local de trabalho, sem que o usuário do sistema faça a transferência de escala e/ou local de trabalho manualmente. O SI deve ainda disponibilizar ao final de cada período de apuração a exportação das horas calculadas no período em forma de evento para o sistema de folha de pagamento da empresa.

O diagrama de contexto do SI implementado é apresentado na Figura 1 a seguir:

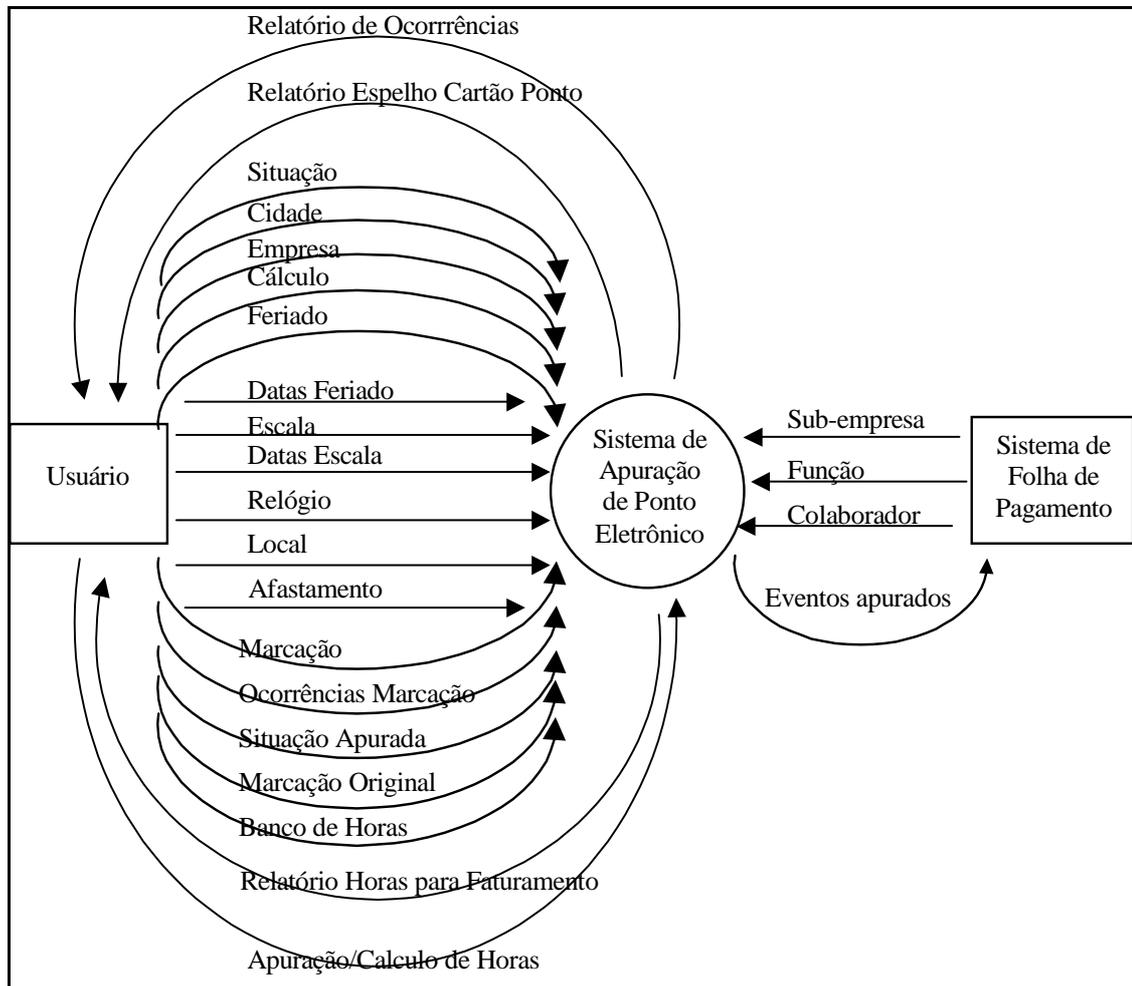


Figura 1 – Diagrama de contexto do SI

### 3.2.3 Diagrama de Fluxo de Dados

Os Diagramas de Fluxo de Dados (DFD) do SI implementado são apresentados a seguir.

Evento 1: conforme a Figura 2, o usuário do sistema faz manutenção (inclusão, alteração, exclusão) nos atributos de Cidade que são armazenados no repositório de dados denominado Cidade.

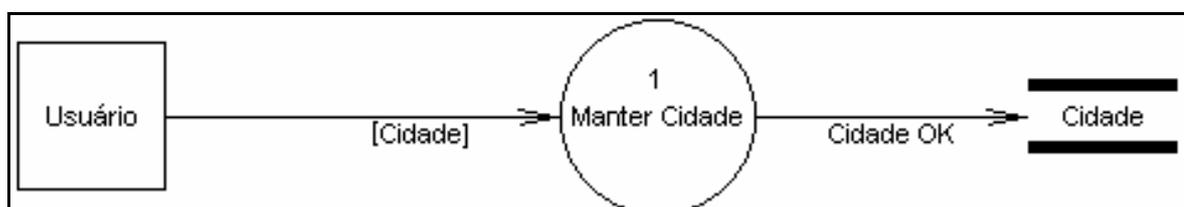


Figura 2 – DFD – Usuário mantém Cidade

Evento 2: na Figura 3, o usuário mantém (inclui, exclui, altera) os atributos da Empresa que faz referência a entidade Cidade e armazena as informações no repositório de dados Empresa.

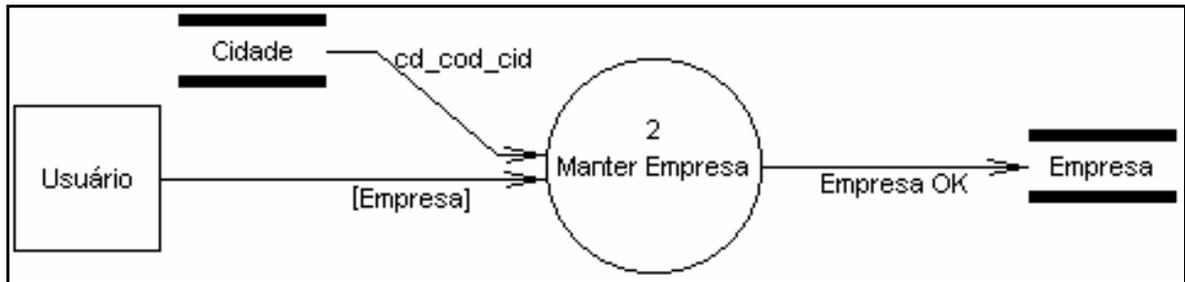


Figura 3 – DFD – Usuário mantém Empresa

Evento 3: o usuário faz a manutenção dos atributos de Situação conforme a Figura 4. A entidade Situação faz referência a entidade Empresa. Os dados são armazenados no repositório Situação.

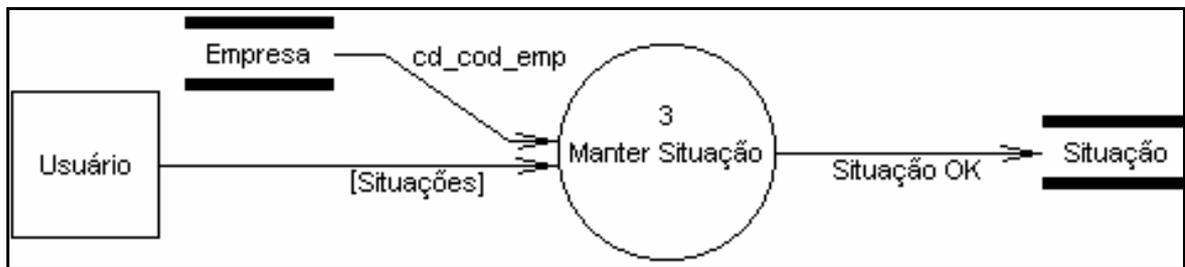


Figura 4 – DFD – Usuário mantém Situação

Evento 4: na Figura 5, o usuário mantém os atributos do Período. O Período contém informação referente ao período de apuração e ao período de cálculo da folha de pagamento. A entidade Cálculo referencia à entidade Empresa. Os dados são armazenados no repositório denominado Cálculo.

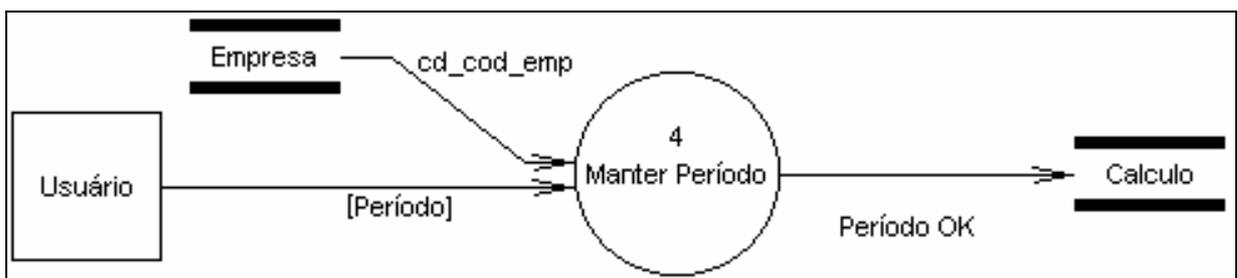


Figura 5 – DFD – Usuário mantém Cálculo

Evento 5: na Figura 6, o usuário do sistema faz a manutenção dos atributos de Feriado que são armazenados na repositório de dados Feriado.

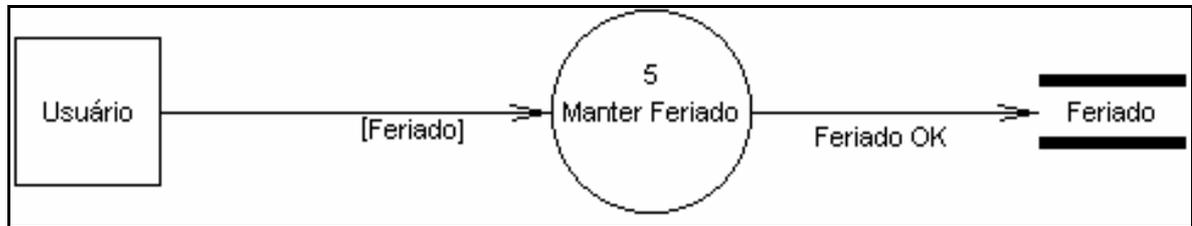


Figura 6 – DFD – Usuário mantém Feriado

Evento 6: o usuário mantém os atributos de Datas Feriado que faz referência a entidade Feriado conforme Figura 7. Para manter Datas feriado a entidade Feriado deve estar previamente mantenedida. O repositório de dados é Datas Feriado.

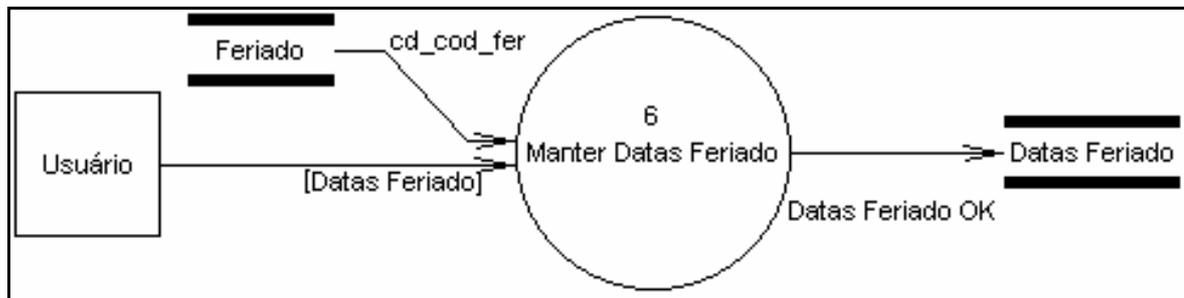


Figura 7 – DFD – Usuário mantém Datas Feriado

Evento 7: na Figura 8, o usuário do sistema faz a manutenção dos atributos da Escala que são armazenados no repositório de dados Escala. A entidade Escala faz referência à entidade Feriado.

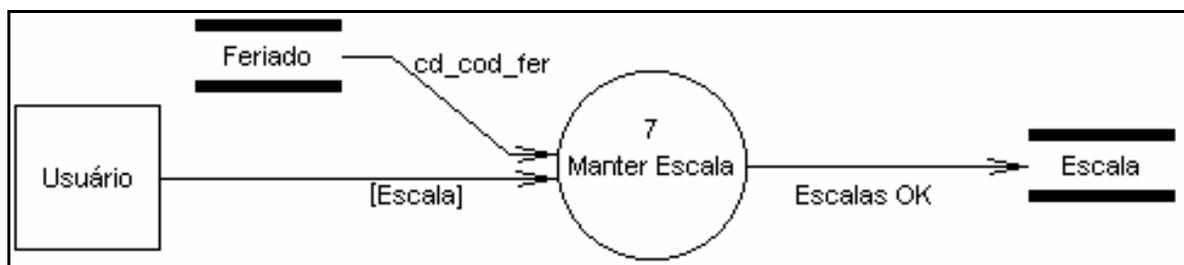


Figura 8 – DFD – Usuário mantém Escala

Evento 8: o usuário mantém Datas Escala conforme Figura 9. Para manter os atributos em Datas Escala a entidade Escala deve estar previamente mantenedida. Os dados são armazenados no repositório Datas Escala.

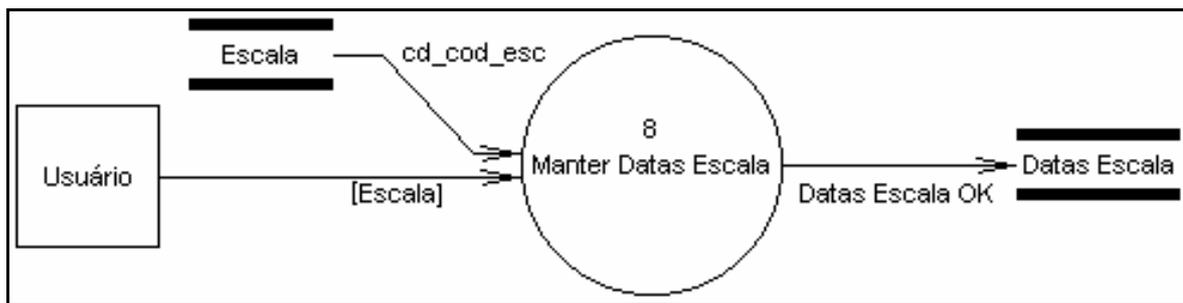


Figura 9 – DFD – Usuário mantém Datas Escala

Evento 9: na Figura 10, o usuário do sistema faz manutenção em Relógio/Coletor. A entidade Relógio faz referência a entidade Escala e Feriado. Os dados são armazenados no repositório Relógio.

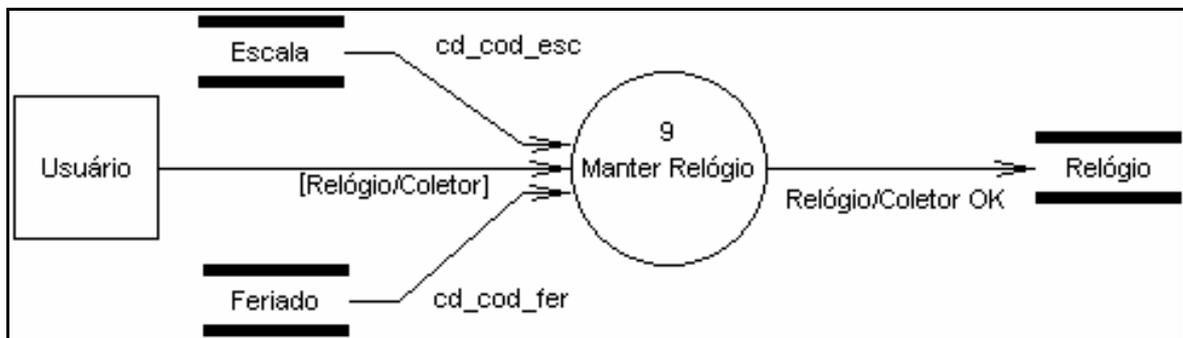


Figura 10 – DFD – Usuário mantém Relógio

Evento 10: na Figura 11, o usuário do sistema faz manutenção em Local. Este referencia as entidades Escala e Feriado. Os dados são armazenados no repositório Local.

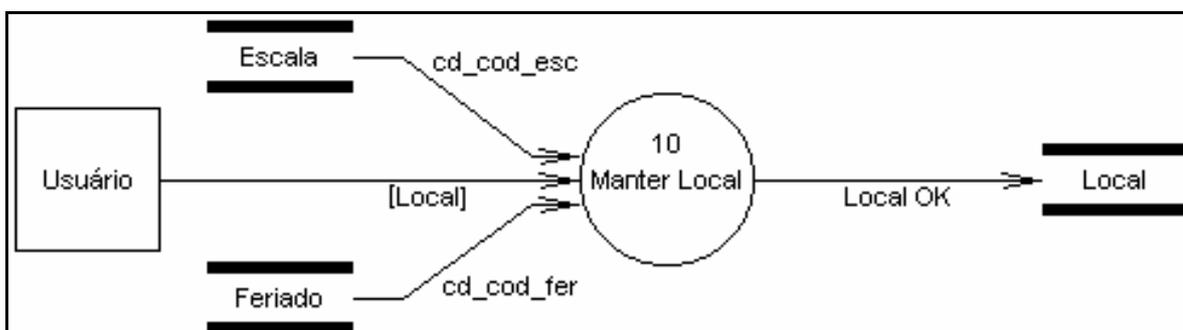


Figura 11 – DFD – Usuário mantém Local

Evento 11: na Figura 12, o usuário faz a manutenção do Afastamento dos colaboradores cadastrados no sistema. A entidade Afastamento faz referência a entidade Colaborador e Situação. Os dados são armazenados no repositório Afastamento.

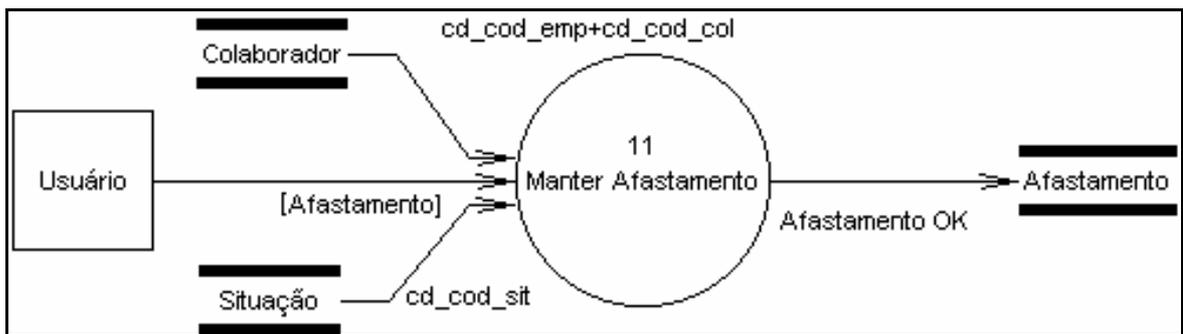


Figura 12 – DFD – Usuário mantém Afastamento

Evento 12: na Figura 13, o usuário do sistema mantém, através da coleta de ponto, a entidade Marcação Original que faz referência as entidades Colaborador, Relógio e Local. As informações são armazenadas no repositório de dados MarOriginal.

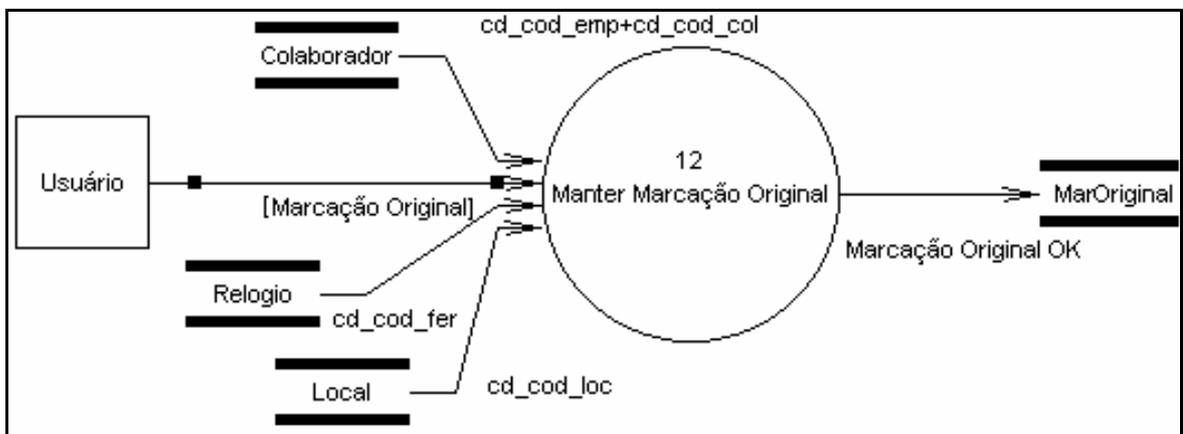


Figura 13 – DFD – Usuário mantém Marcação Original

Evento 13: na Figura 14, o usuário do sistema faz manutenção em Marcação que referência as entidades Colaborador e Cálculo. Os dados são armazenados no repositório Marcação.

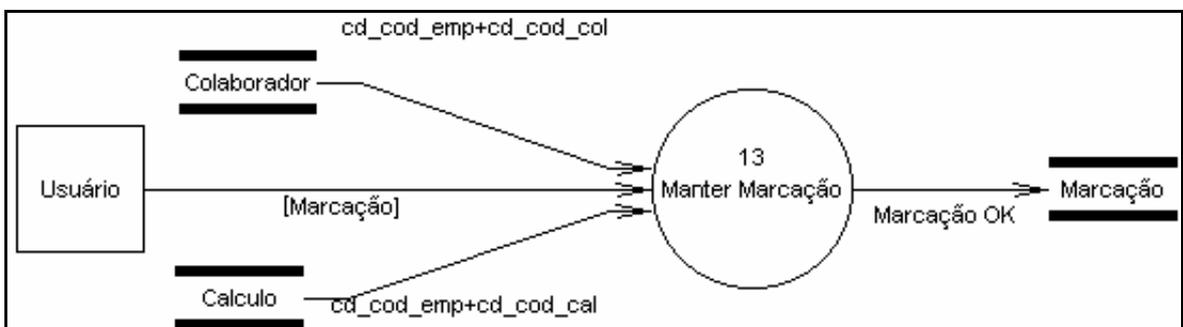


Figura 14 – DFD – Usuário mantém Marcação

Evento 14: na Figura 15, o usuário faz manutenção de Ocorrências Marcação que faz referência a entidade Marcação e Cálculo. Os dados são armazenados no repositório Marocorrenca.

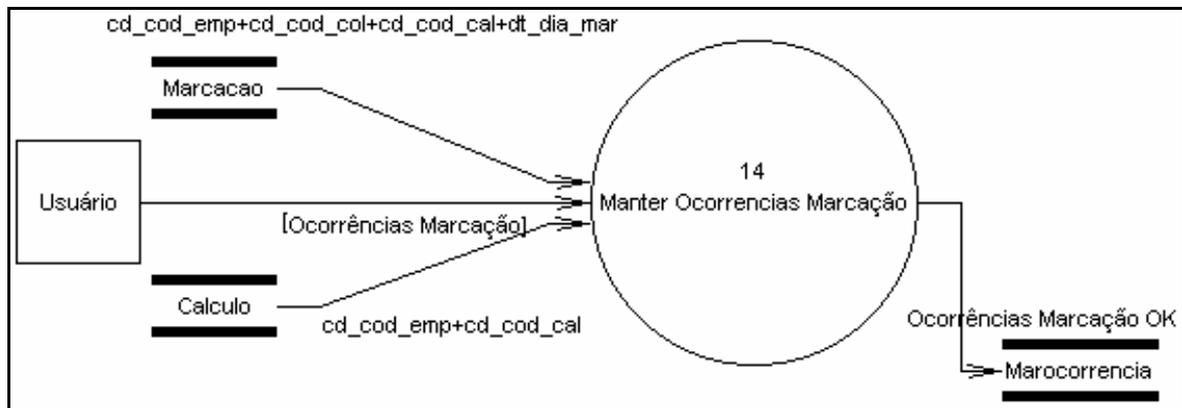


Figura 15 – DFD – Usuário mantém Ocorrências Marcação

Evento 15: na Figura 16, o usuário faz manutenção de Situações Apuradas que faz referência a entidade Marcação e Situação. Os dados são armazenados no repositório Sitapurada.

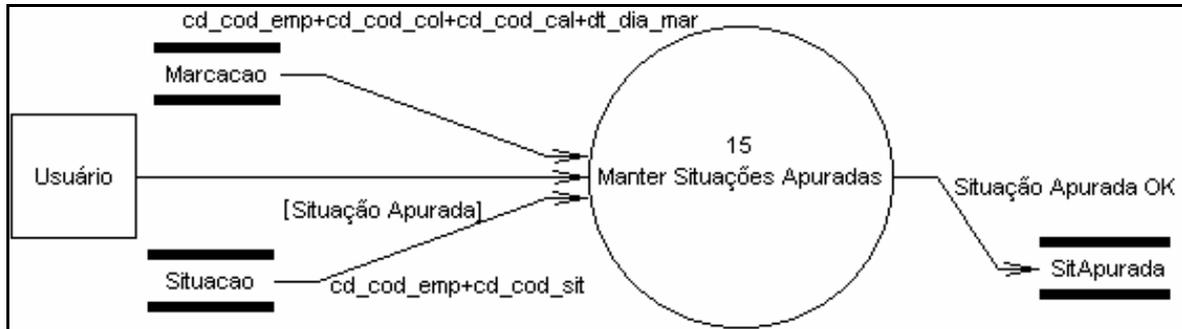


Figura 16 – DFD – Usuário mantém Situação Apurada

Eventos 16: na Figura 17, o usuário mantém os atributos do Banco de Horas que faz referencia as entidades Colaborador, Cálculo, Relógio e Local. Os dados são armazenados no repositório Banco de Hora.

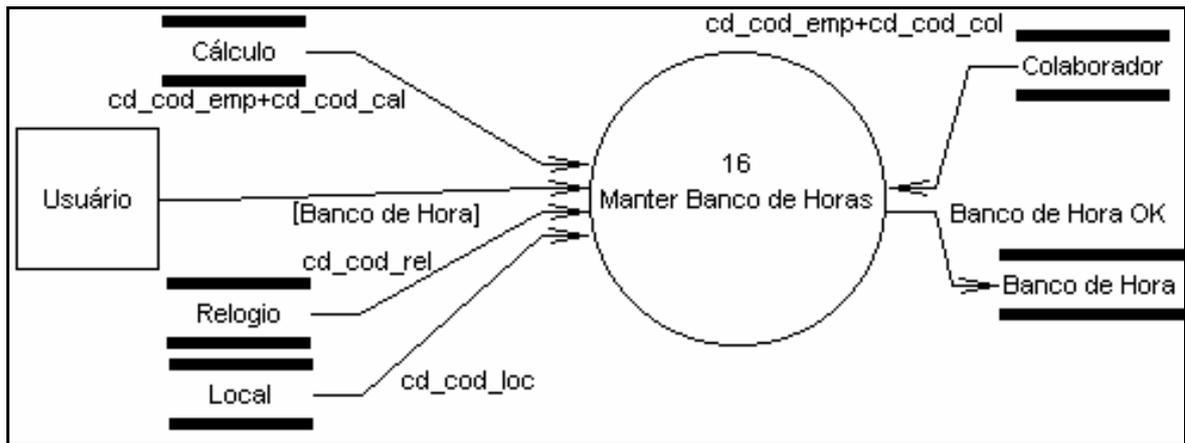


Figura 17 – DFD – Usuário mantém Banco de Horas

Evento 17: o usuário mantém os atributos de Sub-empresa que referênciam à entidade Empresa. Os dados são armazenados na entidade Subempresa conforme mostra a Figura 18.

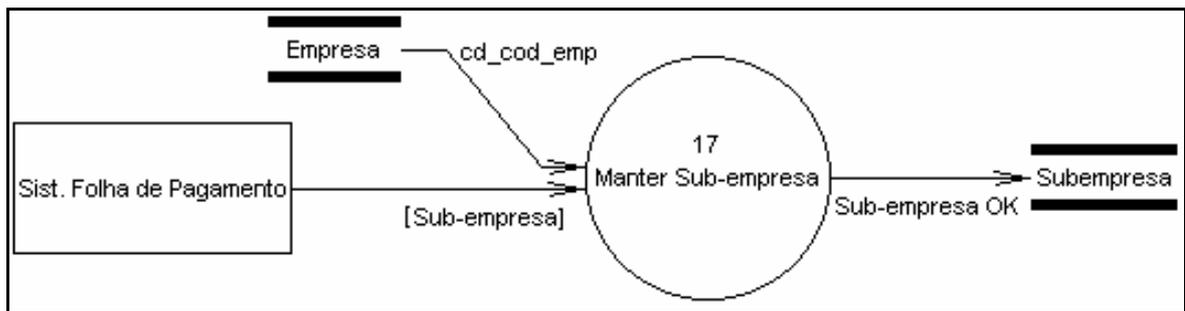


Figura 18 – DFD – Usuário mantém Sub-empresa

Evento 18: o usuário mantém os atributos de Colaborador que referênciam às entidades Empresa, Função e Escala. Os dados são armazenados no repositório Colaborador conforme mostra a Figura 19.

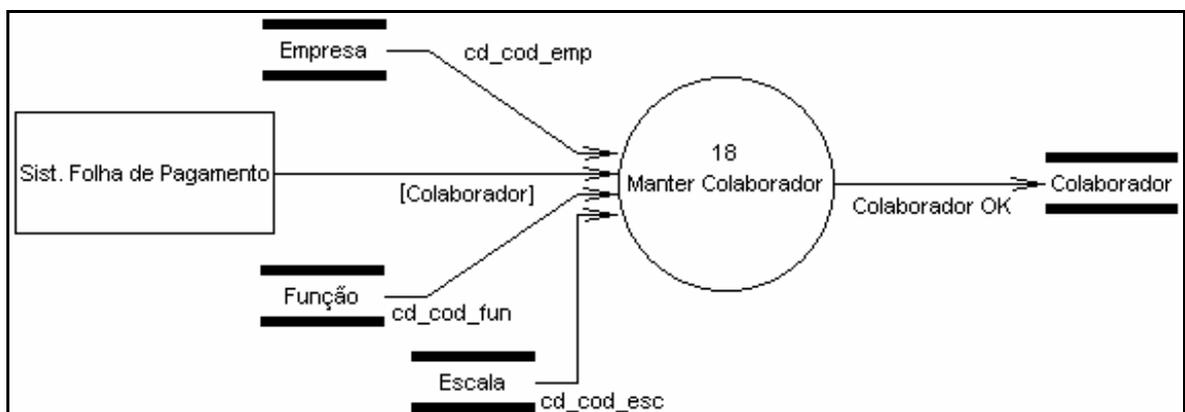


Figura 19 – DFD – Usuário mantém Colaborador

Evento 19: o usuário mantém Função. Os dados são armazenados no repositório Função conforme mostra a Figura 20.

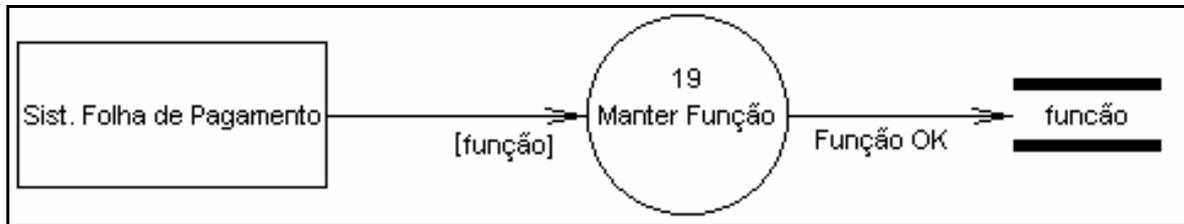


Figura 20 – DFD – Usuário mantém Função

Evento 20: na Figura 21, o usuário emite o relatório situações apuradas que lista os registros da entidade SitApurada e informações complementares das entidades MarOcorrencia, Colaborador, Cálculo, Empresa, Subempresa e Situação. O relatório emite os registros correspondentes a um determinado período informado pelo usuário.

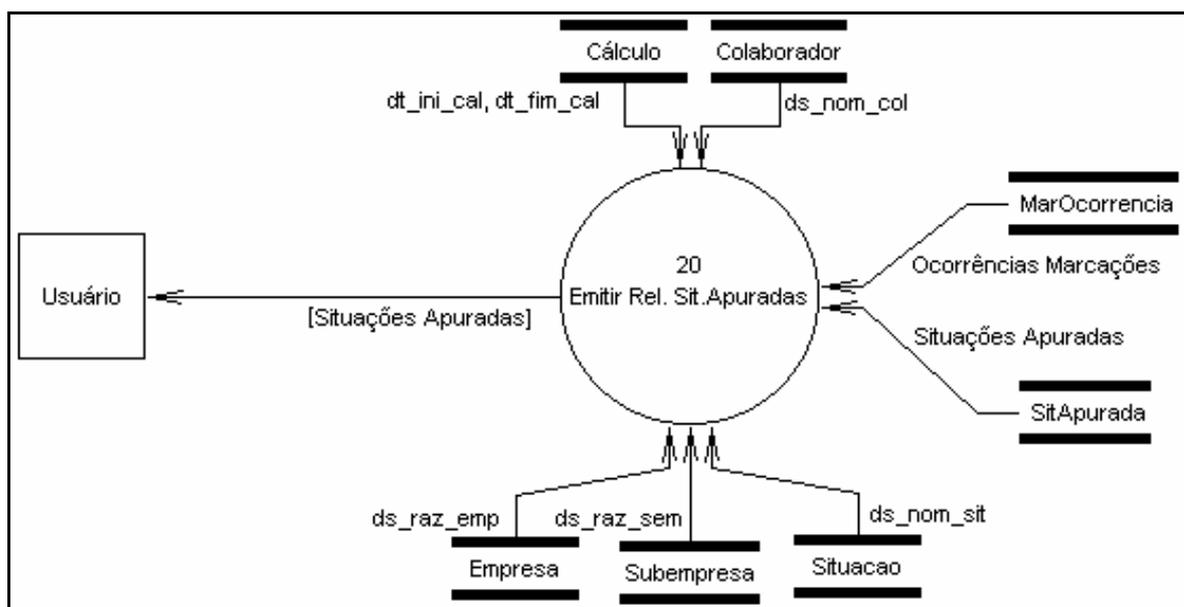


Figura 21 – DFD – Usuário emite Rel. de Situações Apuradas

Evento 21: conforme a Figura 22, o usuário emite o relatório espelho cartão ponto que lista os registros da entidade MarOcorrencia e informações complementares das entidades Colaborador, Cálculo, Empresa e Subempresa. O relatório emite os registros do cartão ponto dos colaboradores referente a um período.

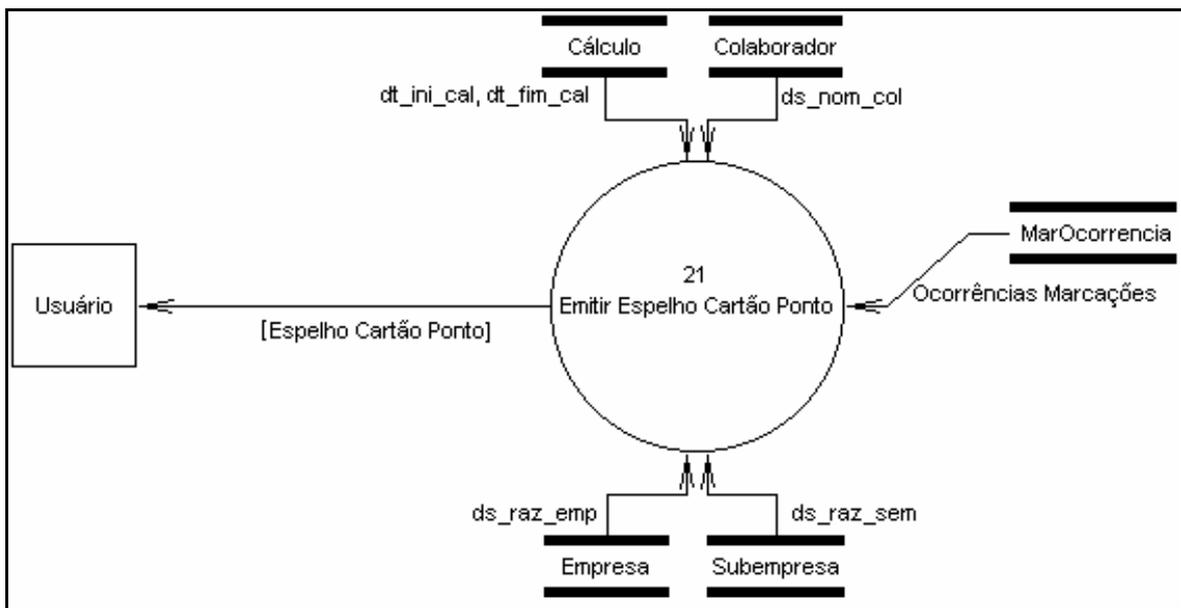


Figura 22 – DFD – Usuário emite Rel. Espelho de Cartão Ponto

Evento 22: conforme a Figura 23, o usuário emite o relatório de horas para faturamento que lista os registros da entidade Sitapurada agrupados por Local (rateio) de trabalho. Informações adicionais são listadas das entidades Colaborador, Cálculo, Função, Subempresa, Empresa e Local.

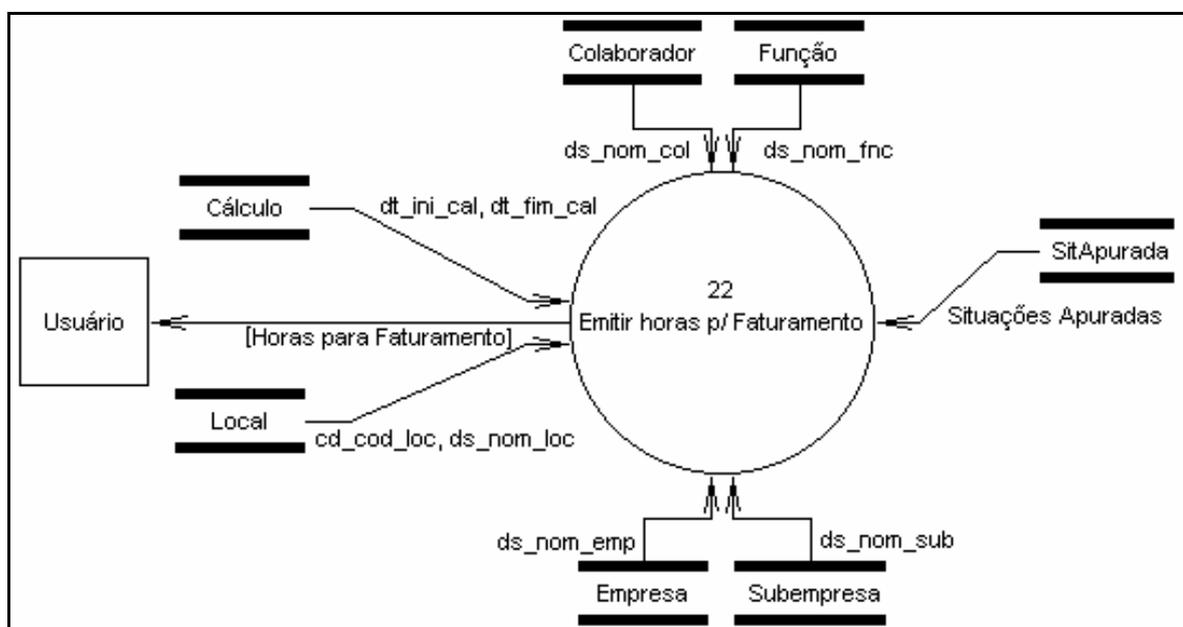


Figura 23 – DFD – Usuário emite Relatório de Horas para Faturamento

Evento 23: este é o principal evento do sistema, que aplica as regras implementadas para o tratamento das marcações efetuadas pelos colaboradores no relógio ponto e/ou coletores de dados. Conforme a Figura 24, o usuário executa a rotina de apuração/cálculo de

horas que lê os registros de um determinado período da entidade Marocorrência (registro de ponto) e das entidades relacionadas Colaborador, Local, Relógio, Escala, Empresa, Cálculo, Situação, Datas Feriado, Datas Escala e Afastamento. Computa as regras implementadas e armazena as situações apuradas no repositório de dados SitApurada e atualiza o repositório Marcação e MarOcorrência.

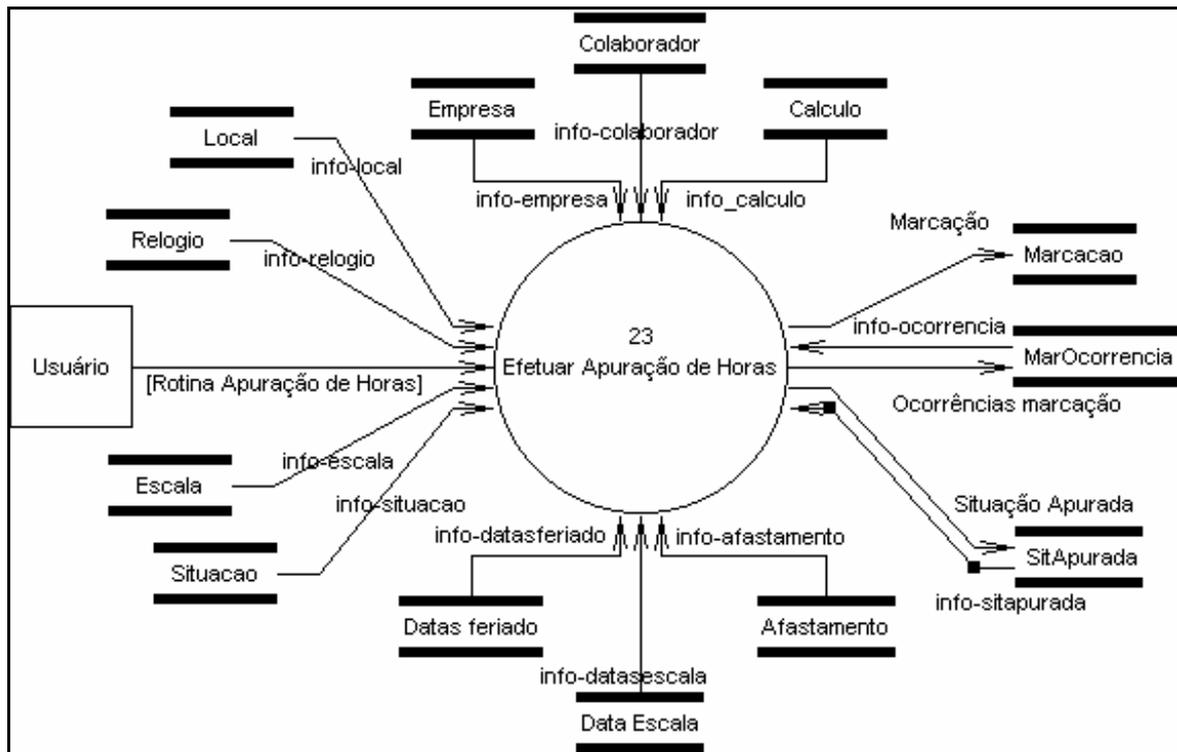


Figura 24 – DFD – usuário executa rotina de apuração/cálculo de horas

### 3.2.4 Diagrama Entidade-Relacionamento – Modelo Conceitual

Para a criação do banco de dados do sistema, foi projetado o Diagrama Entidade-Relacionamento (DER) conceitual, utilizando-se a notação de Peter Chen, apresentado na Figura 25.

O MER resultou em 19 tabelas e 34 relacionamentos. O dicionário de dados das tabelas do banco de dados do sistema implementado pode ser verificado no Apêndice A.

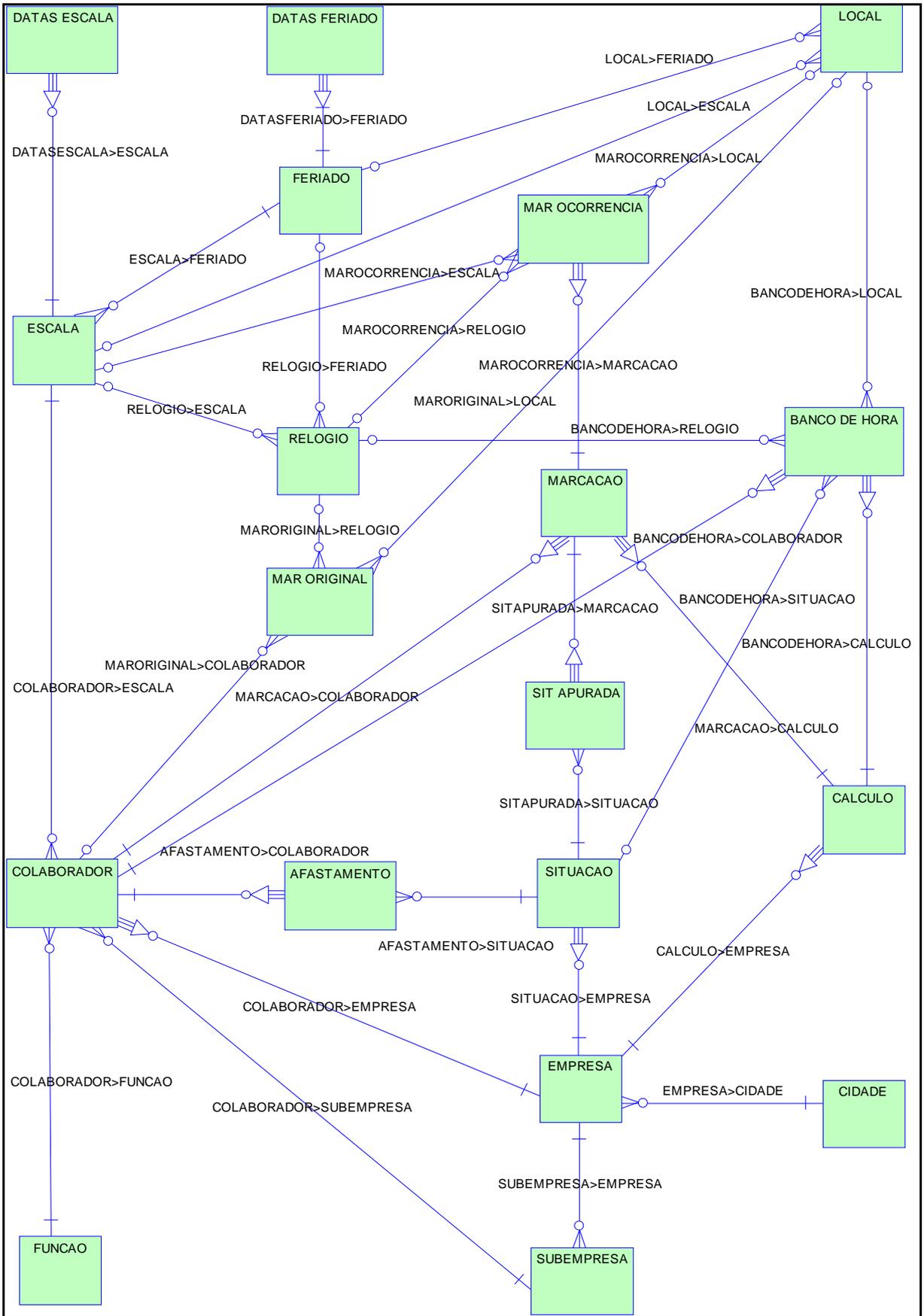


Figura 25 – DER – Modelo Conceitual

3.2.5 Diagrama Entidade-Relacionamento – Modelo Físico

A Figura 26 apresenta o DER Físico do sistema obtido a partir do modelo conceitual apresentado anteriormente.

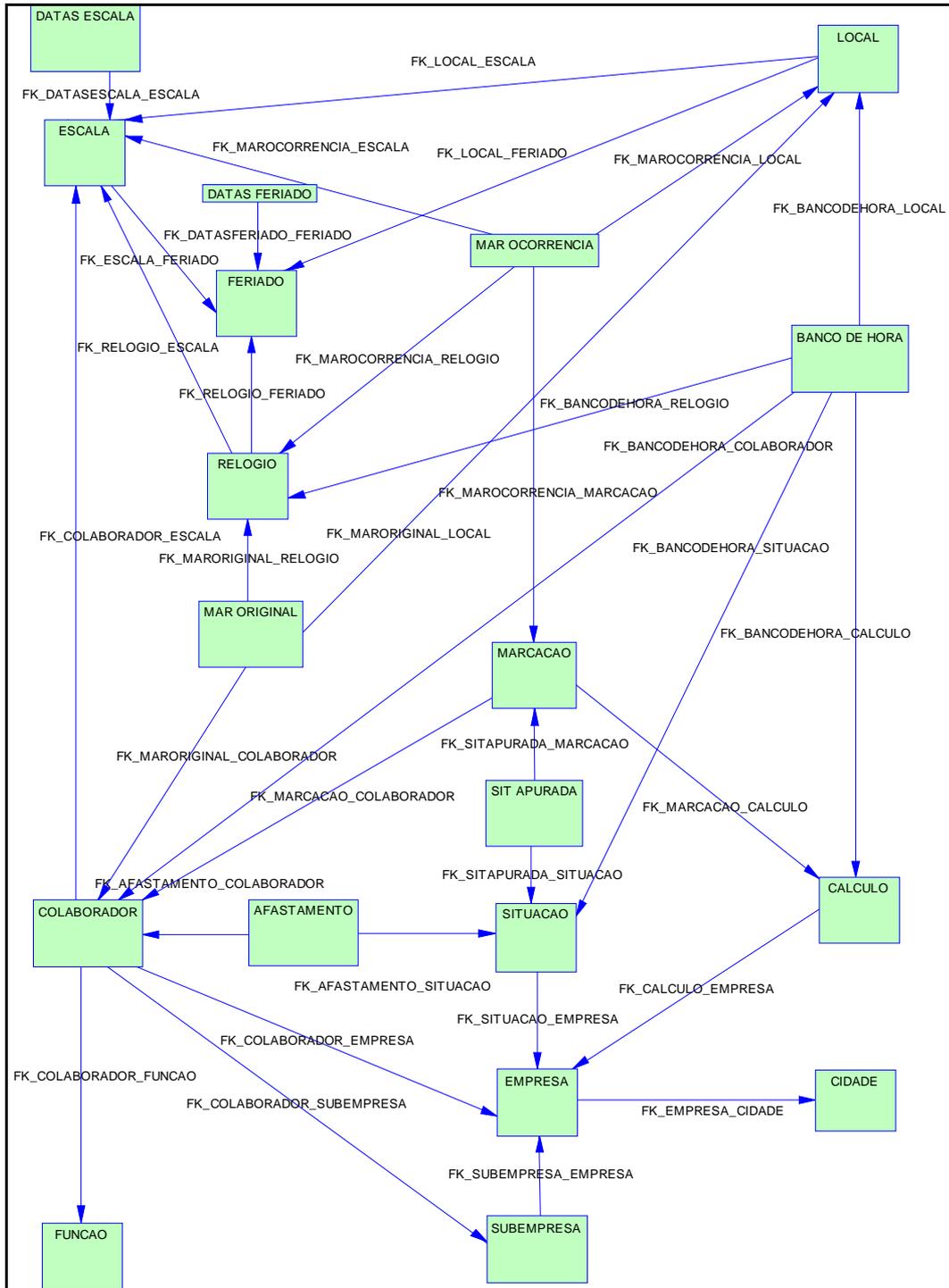


Figura 26 – DER – Modelo Físico

### 3.3 IMPLEMENTAÇÃO

A implementação do SI foi feita com a linguagem de programação *Object Pascal* no ambiente Delphi 7 utilizando a biblioteca de componentes *open source Zeus Database Object* (ZeosDBO) para a conexão com o banco de dados. Para o repositório dos dados do sistema foi utilizado o SGBD PostgreSQL por ser software livre e conforme a revisão bibliográfica ser robusto e confiável. Pelo fato do PostgreSQL não ser nativo ao sistema operacional Windows foi utilizado o emulador Cygwin para a ativação dos processos que permitem a execução do SGBD. O trabalho não prevê a interação do sistema com os dispositivos de coleta de informação (relógio ponto/coletor de dados) e sim com o arquivo texto de dados gerado pelo mesmo.

#### 3.3.1 Técnicas e ferramentas utilizadas

Para a instalação e execução do SGBD PostgreSQL no sistema operacional Windows foi necessário à instalação do emulador de ambiente Unix Cygwin e o Cygipc, um software que gerencia a comunicação em ambiente multitarefa.

Conforme Cygwin (2004c, tradução nossa) o melhor modo recomendado para instalar o Cygwin é utilizando o “Cygwin Setup”. É mais flexível e fácil de usar. Pode-se escolher os pacotes para instalar ou atualizar individualmente. Durante a instalação do Cygwin deve-se selecionar o pacote PostgreSQL para que seja instalado também. Para executar o PostgreSQL é necessária anteriormente à ativação do Cygipc. Instruções detalhadas referentes instalação estão descritas no Apêndice B.

Para a criação do sistema de banco de dados do PostgreSQL foi utilizado o comando “initdb -D /var/pgsql”. Este comando cria um agrupamento de banco de dados no diretório especificado com a opção -D. A ativação do servidor de banco de dados foi feita com o comando “postmaster -i -D /var/pgsql”. O postmaster é o servidor multiusuário do PostgreSQL. A opção -i permite aos clientes se conectarem via TCP/IP no sistema de banco de dados informado na opção -D.

A criação do banco de dados conforme o esquema gerado pelo Power Designer 10 foi feita usando o aplicativo psql do PostgreSQL. O psql é um cliente do PostgreSQL em modo terminal. Permite digitar os comandos interativamente, enviá-los para o PostgreSQL e ver os resultados.

Para o acesso ao banco de dados PostgreSQL a partir do ambiente Delphi foi utilizada a biblioteca de componentes ZeosDBO apresentada na Figura 27. ZeosDBO é um projeto *open source* distribuído sob licença LGPL permitindo assim ser utilizada em software comercial (ZEOS, 2003, tradução nossa).

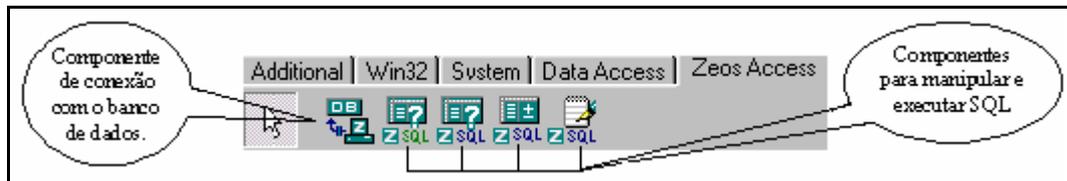


Figura 27 – Paleta de componentes do ZeosDBO

A biblioteca ZeosDBO através dos componentes ZConnection, ZQuery, ZReadOnlyQuery, ZUpdateSQL e ZSQLProcessor permite ao desenvolvedor implementar aplicações com acesso nativo aos servidores de banco de dados *open source* MySQL, PostgreSQL, Firebird e aos comerciais InterBase, MS SQL, Sybase, Oracle e DB2 (ZEOS, 2003, tradução nossa).

O Quadro 1 apresenta o código do método que implementa a conexão com o banco de dados PostgreSQL utilizando o componente de conexão ZConnection da biblioteca ZeosDBO.

```

procedure TfrLogin.pEfetuaConexao;
begin
  { seta as propriedades do objeto de conexao ZeosDBO (TZConnection) }
  dm.conexao.Protocol := 'postgresql-7.2'; //protocolo
  dm.conexao.Port := edPort.GetValue; //porta
  dm.conexao.HostName := edServer.Text; //endereço ip do servidor
  dm.conexao.Database := edDatabase.Text; //nome do banco de dados
  dm.conexao.User := edUser.Text; //usuário
  dm.conexao.Password := edPassword.Text; //senha
  try
    //tenta conexao
    dm.conexao.Connect
  except
    //se houve erro na conexao
    on E : EZSQLException do
      begin
        //tratar a exceção
        pTrataErroConexao(E.ErrorCode);
      end;
    end;
  end;
end;

```

Quadro 1 – Método pEfetuaConexao da classe TfrLogin

Para a implementação das interfaces do sistema foram utilizadas técnicas de programação orientada a objeto (OOP). Os *forms* para manutenção das entidades herdam as características da classe TfrEscolhe.

A classe TfrEscolhe implementa os métodos pInsertCommit, pUpdateCommit e pDeleteCommit que manipulam objetos herdados da classe ZQuery (ZeosDBO) para executar as instruções SQL insert, update, delete.

No Apêndice C é apresentado o método pInsertCommit que é utilizado para executar a instrução SQL insert em qualquer entidade do sistema. Os métodos pUpdateCommit que executa a instrução SQL update e pDeleteCommit que executa a instrução SQL delete tem basicamente a mesma estrutura.

Na classe TfrEscolhe ainda são implementados os métodos pFillFields (Apêndice D) e pClearFields. O método pFillFields preenche o conteúdo dos objetos de edição do *form* correspondente aos atributos da entidade do banco de dados que o usuário seleciona para fazer alteração ou exclusão. O método pClearFields limpa o conteúdo dos objetos de edição cada vez que o usuário seleciona a opção inclusão um novo registro na entidade.

A principal funcionalidade do sistema, a apuração do ponto dos colaboradores, foi elaborada a partir de várias regras definidas juntamente com a empresa. As regras elaboradas estão implementadas na *function* fApuraSituacoes explicada resumidamente a seguir, por conter mais de 1700 linhas de código.

Para isto implementou-se na *unit* uApuraSituacoes as *procedures* pAddSitRat, pSubSitRat, pReplSitRat e pDelSitRat que manipulam as situações rateadas durante a apuração, a *functions* fCalcNot que calcula situações de horas noturnas e fTemAfast que verifica se o colaborador tem afastamento previsto.

No Quadro 2 é apresentado um trecho do código fonte onde estão implementadas algumas regras que apontam as situações.

```

...
////////////////////////////////////
// 01/02-HORAS TRABALHADAS DIURNAS E NOTURNAS
////////////////////////////////////
qtdnot := CalcNot(BATIDAS[J-1].HrMin,BATIDAS[J].HrMin);
SITAPU[02] := SITAPU[02]+qtdnot;
AddSitRat(02,qtdnot,RATEIOS[J-1],RATEIOS[J]);

qtdmin := ((BATIDAS[J].HrMin-BATIDAS[J-1].HrMin) - qtdnot);
SITAPU[01] := SITAPU[01]+qtdmin;
AddSitRat(01,qtdmin,RATEIOS[J-1],RATEIOS[J]);

////////////////////////////////////
// 03/04-EXTRA ANTES DO HORARIO
////////////////////////////////////
if (BATIDAS[J-1].HrMin < hrent01) and (BATIDAS[J].HrMin < hrent01)
begin
qtdnot := CalcNot(BATIDAS[J-1].HrMin,BATIDAS[J].HrMin);
SITAPU[03] := SITAPU[03]+qtdnot;
AddSitRat(03,qtdnot,RATEIOS[J-1],RATEIOS[J]);
//
qtdmin := ((BATIDAS[J].HrMin-BATIDAS[J-1].HrMin) - qtdnot);
SITAPU[04] := SITAPU[04]+qtdmin;
AddSitRat(04,qtdmin,RATEIOS[J-1],RATEIOS[J]);
end;

////////////////////////////////////
// 19/20-EXTRA APOS DO HORARIO
////////////////////////////////////
if (BATIDAS[J-1].HrMin > hrsai02) and (BATIDAS[J].HrMin > hrsai02) then
begin
qtdnot := CalcNot(BATIDAS[J-1].HrMin,BATIDAS[J].HrMin);
SITAPU[20] := SITAPU[20]+qtdnot;
AddSitRat(20,qtdnot,RATEIOS[J-1],RATEIOS[J]);
//
qtdmin := ((BATIDAS[J].HrMin-BATIDAS[J-1].HrMin) - qtdnot);
SITAPU[19] := SITAPU[19]+qtdmin;
AddSitRat(19,qtdmin,RATEIOS[J-1],RATEIOS[J]);
end;
//trata situacoes de extras no primeiro periodo
if (BATIDAS[J-1].HrMin >= hrent01) or (BATIDAS[J].HrMin >= hrent01) then
begin
////////////////////////////////////
// 05/06-EXTRA NA ENTRADA (1º Período)
////////////////////////////////////
qtdnot := 0;
qtdmin := hrent01-BATIDAS[J-1].HrMin; //checa pela entrada
//se tem extra
if qtdmin >= 0 then
begin
// se tem extra
if qtdmin > 0 then
begin
qtdnot := CalcNot(BATIDAS[J-1].HrMin,hrent01);
if qtdnot < 0 then
begin
qtdnot := 0;
end;
end;
//se primeira vez, assume qtdmin como menor extra
qtddiu := (qtdmin-qtdnot);
if PriExE1 then
begin
SITAPU[05] := qtddiu;
ReplSitRat(05,qtddiu,RATEIOS[J-1],RATEIOS[J-1]);
SITAPU[06] := qtdnot;
ReplSitRat(06,qtdnot,RATEIOS[J-1],RATEIOS[J-1]);
PriExE1 := False;
end
//senao, assumimos qtdmin se for menor que tempo anterior
else
if (qtddiu < SITAPU[05]) then
begin
SITAPU[05] := qtddiu;
ReplSitRat(05,qtddiu,RATEIOS[J-1],RATEIOS[J-1]);
SITAPU[06] := qtdnot;
ReplSitRat(06,qtdnot,RATEIOS[J-1],RATEIOS[J-1]);
end;
end;
end;
end;
...

```

Calcula a quantidade de horas para a situação 01-Horas trabalhadas diurnas e 02-Horas trabalhadas noturnas

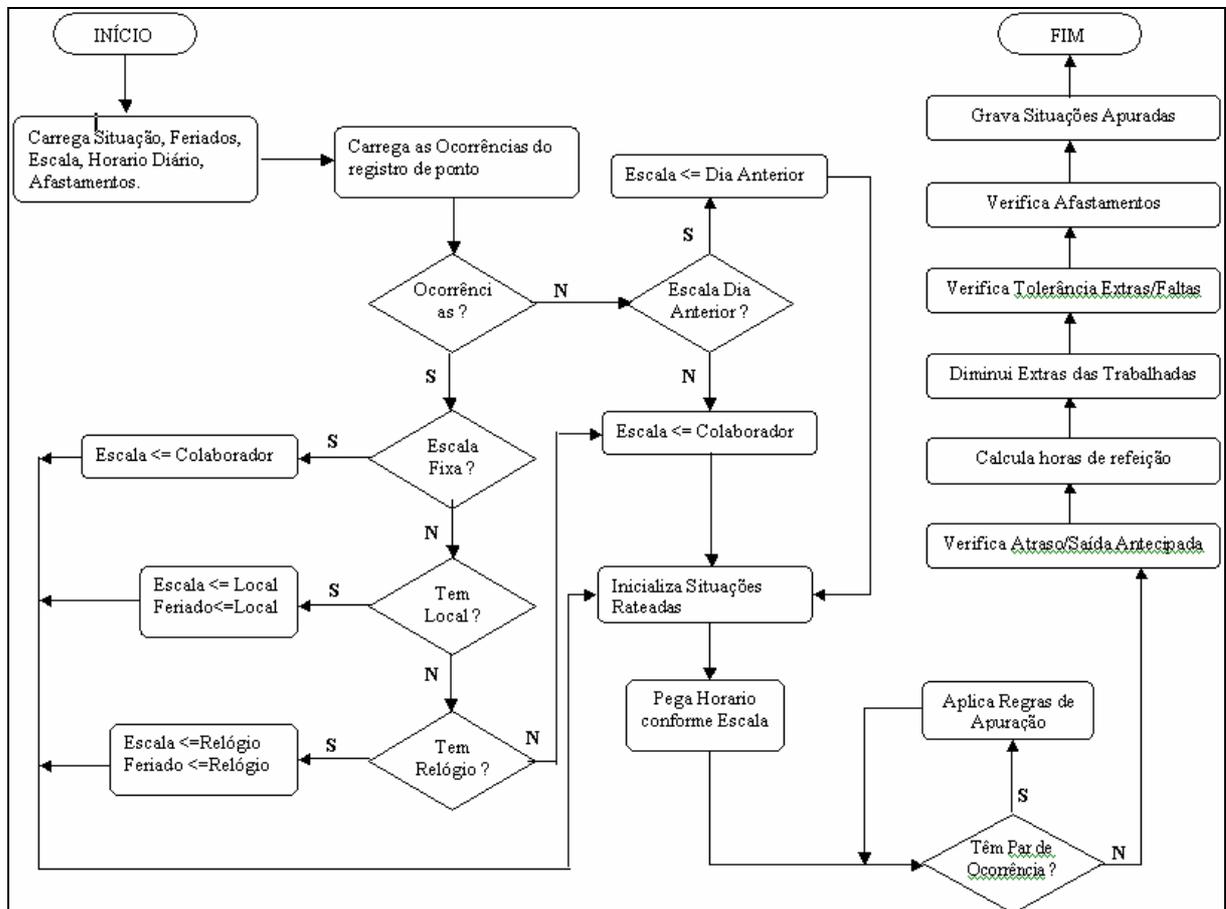
Calcula a quantidade de horas extras para situação 03-Extras Antes Noturnas e 04-Extras Antes Diurnas

Calcula a quantidade de horas extras para situação 20-Extras após Noturnas e 19-Extras após Diurnas

Calcula a quantidade de horas extras para situação 05-Extras entrada 1º parte diurna e 06-Extras entrada 1º parte noturna

Quadro 2 – Trecho da *function* fApuraSituacao

A lógica do algoritmo para apontamento das situações é simplificada e apresentada no fluxograma do Quadro 3.



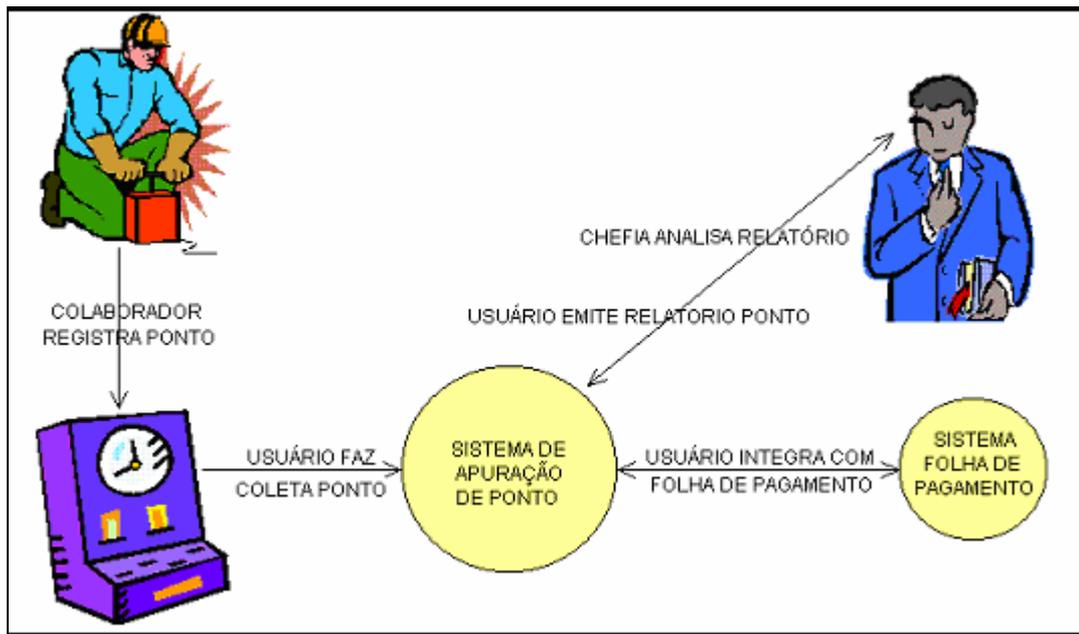
Quadro 3 – Fluxograma - Algoritmo de Apuração

No processo “Aplica Regras de Apuração”, para cada par de registros de ponto o sistema compara com o horário da escala e determina o melhor resultado para a situação, tais como: trabalho diurno, trabalho noturno, horas extras, atraso, saída antecipada. A relação completa das situações que o sistema trata no algoritmo encontra-se no Apêndice E.

As situações apuradas são rateadas, ou seja, quando um colaborador registrar o ponto em locais diferente num mesmo dia, as situações são calculadas proporcionalmente para cada local registrado.

### 3.3.2 Operacionalidade da implementação

O Quadro 3 apresenta um diagrama que mostra a interação entre os agentes externos e o sistema desenvolvido.



Quadro 4 – Diagrama de Interação do Sistema

O Sistema permite trabalhar com múltiplas empresas e cada empresa permite operar com diferentes períodos de apuração de ponto. Entende-se como período de apuração, por exemplo, trinta dias de um determinado mês.

A empresa a operar e o período de apuração devem ser selecionados na opção “Trocar Empresa/Período” no menu “Ponto”. A Figura 28 mostra a interface para seleção.



Figura 28 – Interface Seleciona Empresa/Período de Apuração

A empresa e o período selecionado podem ser visualizados na barra de *status* do sistema localizada no rodapé da interface principal (Figura 29).



Figura 29 – Barra de *Status* do Sistema

Ao acessar as interfaces de manutenção das entidades do sistema os registros são apresentados em uma grade no modo “Consulta”. Todas seguem o padrão apresentado na Figura 30.

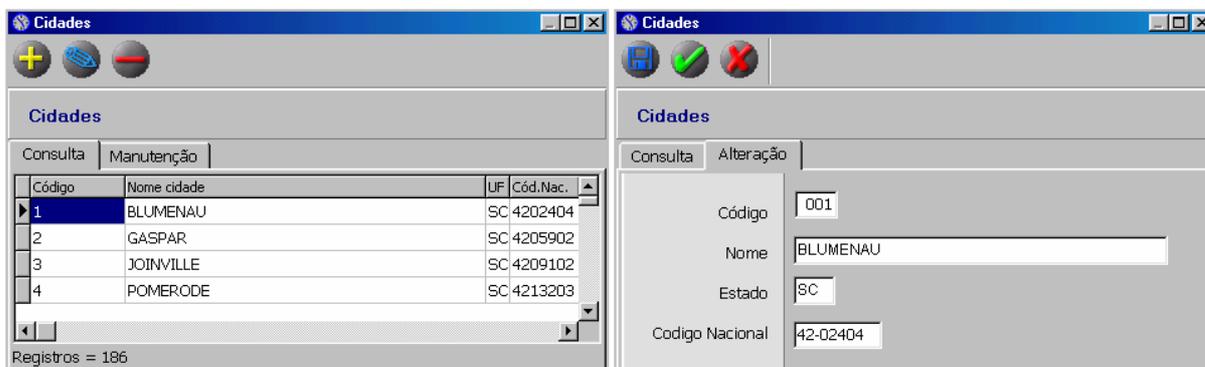


Figura 30 – Interface Cidades

No modo “Consulta”, clicando os botões “Incluir” , “Alterar”  e “Excluir”  ou pressionando as teclas “Insert”, “Enter” e “Delete” é possível fazer a inclusão, alteração ou exclusão dos registros respectivamente.

No modo de “Manutenção” clicando o botão “Gravar”  é efetuada a gravação do registro no banco de dados. O botão “Confirmar”  ou a tecla “PageDown”, gravam o registro e retorna ao modo de “Consulta”. E o botão “Cancelar”  ou a tecla “Escape” retornam a “Consulta” sem gravar o registro.

Nas páginas a seguir são apresentadas as entidades básicas do sistema.

Inicialmente deve-se cadastrar os dados das entidades escala, feriado, relógio, local, cidade e função através das respectivas interfaces localizadas no menu “Tabelas”. Os registros destas entidades são comuns a qualquer empresa que o usuário estiver operando.

As entidades colaborador, sub-empresa e situação são subordinadas a empresa que está selecionada na barra de *status* do sistema.

As entidades colaborador, sub-empresa e função devem preferencialmente ser atualizadas através da opção “Importar Colaboradores...” no menu “Integração”. Assim evita-se erro nas informações destas entidades comuns ao sistema de ponto e o sistema de folha de pagamento da empresa. Se o usuário preferir poderá manter os registros manualmente.

A seguir são apresentadas as interfaces que necessitam de atenção especial para o correto funcionamento da apuração do registro de ponto.

A interface de manutenção das escalas apresentada na Figura 31, possibilita o cadastramento de diferentes escalas de horário. As escalas de horário são utilizadas nas interfaces locais, relógios, colaboradores e ocorrências do ponto.

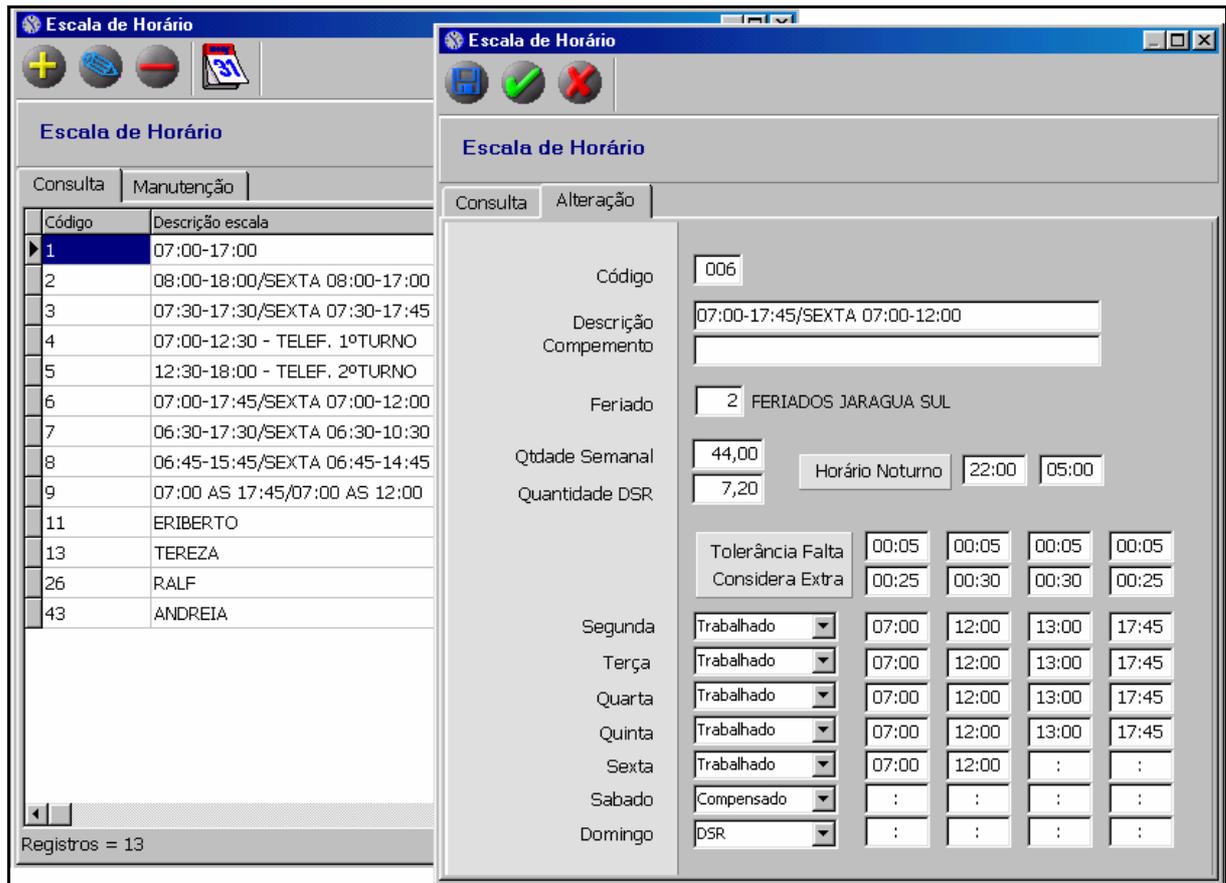


Figura 31 – Interface Escalas de Horário

A seguir é apresentada a descrição dos atributos mais importantes desta interface:

- feriado: informe o código do feriados que será considerada para esta escala na apuração em dias de feriado;
- quantidade semanal: informe a quantidade de horas correspondente à carga horária de trabalho semanal;
- quantidade DSR: informe a quantidade de horas correspondente ao Descanso Semanal Remunerado (DSR).
- horário noturno: Informe o horário inicial e o horário final que deve ser considerado como horas noturnas durante a apuração;

- e) tolerância falta: informe a quantidade mínima de minutos que o sistema deve considerar como de atraso ou saída antecipada. Situações apuradas com quantidade inferior a quantidade informada não serão geradas na apuração;
- f) considera extra: informe a quantidade de mínima de minutos que o sistema deve considerar como extra. Situações apuradas com quantidade inferior a quantidade informada não serão geradas na apuração;
- g) segunda, terça, quarta, quinta, sexta, sábado, domingo (tipo dia): para cada dia da semana deve-se informar o tipo do dia para considerar na apuração. Pode-se, por exemplo informar “Trabalhado” em dias úteis; “Compensado” para sábado; “DSR” para Domingo;
- h) segunda, terça, quarta, quinta, sexta, sábado, domingo (horários): para cada dia da semana deve-se informar a hora em que o colaborador deverá registrar o seu ponto. Os horários diários estão divididos em duas partes. Na primeira parte tem-se a hora de entrada e a hora de saída que o colaborador deverá registrar antes da refeição. Na segunda parte tem-se a hora de entrada e hora de saída que o colaborador deverá registrar após a refeição. Obs: caso a escala não tenha intervalo para refeição deve-se cadastrar apenas os dois primeiros campos correspondentes à hora de entrada e a hora de saída.

Para uma flexibilização das escalas de horário é possível cadastrar um horário diferente em uma data específica, clicando-se no ícone  localizado na interface “Escala de Horário” no modo “Consulta” conforme Figura 31.

A manutenção dos locais de trabalho é feita na interface “Locais” como mostra a Figura 32. O rateio das horas trabalhadas para um determinado local é calculado quando houver um local registrado em uma ocorrência no registro de ponto conforme pode-se observar na Figura 37.

The image shows a software window titled "Local" with a menu bar containing "Consulta" and "Alteração". The main content area is a form with the following fields:

- Código:** 02381E
- Descrição:** EMPREITADA 02381-E
- Escala:** 6 - 07:00-17:45/SEXTA 07:00-12:00
- Feriado:** 3 - FERIADO STA CRUZ/RS
- Rateio:** 0876
- Horário Noturno:** Two empty input fields separated by a colon (:).
- Data atualização:** 26/08/2003

Figura 32 – Interface Locais

A seguir são descritos os principais atributos da interface “Local”:

- escala: O atributo escala deve ser informado para haver a troca automática de escala. Assim, quando um colaborador registrar o ponto neste local, durante a apuração serão calculadas as horas considerando a escala informada;
- feriado: informe caso o local deve seguir um feriado específico para considerar na apuração de ponto;
- rateio: informe o código do rateio que deve gerar para as situações apuradas neste local. Este campo é gravado no arquivo gerado na opção “Exportar eventos para folha...” no menu “Integração”;
- horário Noturno: Informe o horário inicial e o horário final que deve ser considerado como horas noturnas durante a apuração, caso seja diferente do informado na escala.

A interface “Relógios/Coletores” possibilita cadastrar os relógios de ponto e/ou coletores de dados utilizados para registrar o ponto. Seus atributos têm as mesmas características dos atributos da interface “Local”.

Os locais e/ou relógios são incluídos automaticamente quando surgir um novo código no arquivo lido durante a execução da opção “Coletar Ponto...” no menu “Ponto”. O usuário deve então acessar as respectivas interfaces para fazer a manutenção destes novos registros.

Conforme a Figura 33, a interface de manutenção de situações possibilita a parametrização das 48 situações (Apêndice E) pré-definidas que são tratadas na apuração do ponto.

Figura 33 – Interface Situações

A seguir são descritos os atributos que compõem a interface de manutenção das situações:

- a) código: código numérico interno do sistema (de 1 a 48);
- b) descrição: descrição interna da situação. Não pode ser alterada;
- c) permite justificativa/abono: informe se a situação quando apurada poderá ser abonada ou justificada, evitando assim que a perda das horas apuradas em situações como falta, atraso, saída antecipada;
- d) considerar como exceção: informe se a situação quando apurada, deverá ou não aparecer nos relatórios;
- e) tratar na apuração: informe se a situação deve ou não ser calculada na apuração;
- f) evento folha período atual: informe o código do evento no sistema de folha de pagamento correspondente à situação apurada nos dias compreendidos entre a data inicial da folha e a data final da apuração. Este campo é gravado no arquivo gerado na opção “Exportar eventos folha” no menu “Integração”;
- g) evento folha período anterior: informe o código do evento no sistema de folha de pagamento correspondente à situação apurada nos dias compreendidos entre a data inicial da apuração e o dia anterior à data inicial da folha. Este campo é gravado no arquivo gerado na opção “Exportar eventos folha” no menu “Integração”;

- h) considerar como (banco de horas): informe se a situação quando apurada deverá creditar ou debitar as horas no banco de horas. Para as situações de horas extras deve-se creditar e para faltas, atraso, saída antecipada deve-se debitar. O sistema só irá gerar um lançamento no banco de horas se o colaborador for optante do banco de horas;
- i) multiplicador (banco de horas): informe neste campo o valor pelo qual as horas apuradas na situação deverão ser multiplicadas para creditar ou debitar no banco de horas;
- j) evento de folha (banco de horas): informe o código do evento no sistema de folha de pagamento correspondente ao débito das horas apurada do banco de horas nos dias compreendidos entre a data inicial da folha e a data final da apuração. Este campo é gravado no arquivo gerado na opção “Exportar eventos folha” no menu “Integração”.

A interface “Situações”, diferentemente da outras, não permite a inclusão ou exclusão dos registros pelo usuário do sistema. Só é permitida a alteração. Estes registros são necessários para a correta apuração diária do ponto dos colaboradores. No término de um período é possível a exportação das horas calculadas na forma de eventos para o sistema de folha de pagamento.

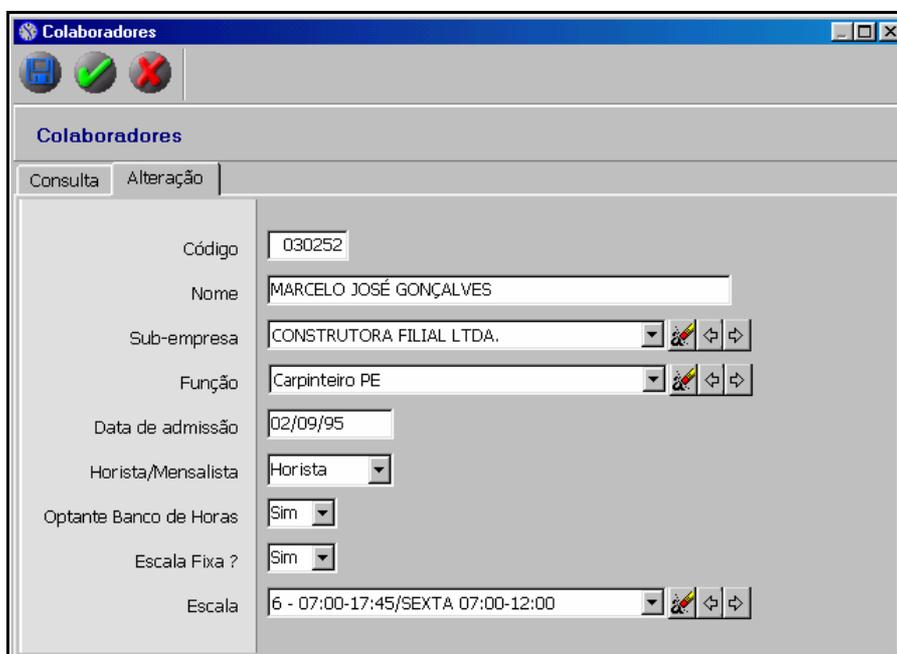
Na opção “Empresa...”, deve-se cadastrar as empresas cujos colaboradores terão seu registro de ponto apurado pelo sistema.

Para cada empresa deve-se cadastrar na opção “Períodos...” os períodos correspondentes ao registro de ponto dos colaboradores para efetuar a apuração.

A seguir estão relacionados os atributos pertinentes a interface “Períodos” :

- a) código: deve-se informar um código numérico para identificar o período de apuração que será calculada as horas dos colaboradores;
- b) período de apuração: informe a data inicial e a data final correspondente ao registro de ponto dos colaboradores a qual o sistema deverá considerar para fazer a apuração;
- c) período de folha: informe a data inicial e a data final do sistema de folha de pagamento correspondente ao período de apuração.

Dentre as principais interfaces do sistema apresenta-se ainda a de “Colaboradores” na Figura 34. Nela Interface deve-se parametrizar os colaboradores que a empresa deseja tratar o registro de ponto.



The screenshot shows a software window titled "Colaboradores" with a menu bar containing "Consulta" and "Alteração". The main area contains a form with the following fields:

- Código: 030252
- Nome: MARCELO JOSÉ GONÇALVES
- Sub-empresa: CONSTRUTORA FILIAL LTDA. (dropdown menu)
- Função: Carpinteiro PE (dropdown menu)
- Data de admissão: 02/09/95
- Horista/Mensalista: Horista (dropdown menu)
- Optante Banco de Horas: Sim (dropdown menu)
- Escala Fixa?: Sim (dropdown menu)
- Escala: 6 - 07:00-17:45/SEXTA 07:00-12:00 (dropdown menu)

Figura 34 – Interface Colaboradores

A seguir são descritos os principais atributos (relevantes à apuração/cálculo de horas) que compõem a interface “Colaboradores” :

- horista/mensalista: informe Horista se o salário do colaborador é calculado baseado nas horas efetivamente trabalhada durante o mês ou informe Mensalista se o salário é baseado em uma quantidade fixa de horas durante mês (referente a 30 dias);
- optante banco de horas: informe “sim” se as ausências e/ou extras apuradas para o colaborador devem ser creditadas e/ou debitadas do banco de horas;
- escala fixa: informe “sim” se o colaborador deve exercer diariamente o horário de trabalho conforme informado no atributo “Escala” independente do local onde trabalhar ou informe “não” se o colaborador deve exercer o horário de trabalho conforme o local e/ou relógio onde registrar o ponto. Na segunda forma o colaborador terá sua transferência automática de escala cada vez que registrar o ponto em um local e/ou relógio diferente do dia anterior.

A seguir é exemplificada a principal funcionalidade do sistema, o apontamento do registro de ponto dos colaboradores. As opções encontram-se no menu “Ponto”.

Tendo selecionado a empresa e o período de apuração, periodicamente deve-se efetuar a coleta dos arquivos provenientes dos relógios eletrônicos ou dos coletores de dados da empresa através da opção “Coletar Ponto...”.

Após a coleta do ponto deve-se efetuar a apuração do ponto para que o sistema calcule as horas diárias das ocorrências de ponto.

A apuração do ponto pode ser feita coletivamente através da opção “Apurar Ponto...” ou individualmente na interface de manutenção do registro de ponto (Figura 37) clicando no ícone  (Apurar). A apuração coletiva do ponto é iniciada após a confirmação dos dados preenchidos na interface apresentada na Figura 35, selecionando-se a opção “Apurar Ponto...”.

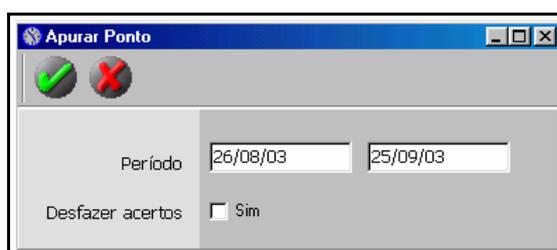


Figura 35 – Interface Apurar Ponto

Deve-se informar a data de início e a data final que o sistema deverá ser considerado para fazer a apuração das horas. Estas devem compreender entre o primeiro e o último dia do período. Verifica-se na barra de *status* do sistema.

Optando-se por “Desfazer acertos” o sistema irá eliminar as situações apuradas anteriormente cujo usuário tenha feito algum acerto manualmente na interface “Situações apuradas”.

Após a apuração do ponto deve-se emitir o “Relatório de Apuração” para conferência e acerto das situações apuradas. A Figura 36 demonstra este relatório.

CONSTRUTORA LTDA.				PERÍODO:148 DE 26/04/2003 A 25/05/2003																				
RELATORIO DE SITUAÇÕES APURADAS																								
Sem	Dt	apurado	Es	Tp	Tipo	Dia	Bat	Hr	Local	Re	Bat	Hr	Local	Re	Bat	Hr	Local	Re	Bat	Hr	Local			
=====19682 JOSE JOAO GARCIA===== Adm:03/02/1986 == Sit:001 em 01/02/2003 =====																								
Sab	26/04/2003	06	C	Compensado			06:46	06	02094-E		17:25	06	02094-E											
							21	Ext	em dia Cmp - Diu		10:39	02094-E		bh(+)										
Seg	28/04/2003	06	T				06:45	06	02094-E		19:19	06	02094-E100											
							17	Extra	Saida	2o.	1:34	02094-E		bh(+)										
Ter	29/04/2003	06	T				06:45	06	02094-E		19:16	06	02094-E											
							17	Extra	Saida	2o.	1:31	02094-E		bh(+)										
Qua	30/04/2003	06	T				06:45	06	02094-E		19:19	06	02094-E100											
							17	Extra	Saida	2o.	1:34	02094-E		bh(+)										
Qui	01/05/2003	06	T				11	Falta	1o. periodo		5:00													
Sex	02/05/2003	06	T				06:46	06	02094-E		18:49	06	02094-E											
							17	Extra	Saida	2o.	6:49	02094-E		bh(+)										

Figura 36 – Relatório de Situações Apuradas

Após a emissão deste relatório deve-se distribuí-lo para os chefes de cada local de trabalho da empresa. Estes devem fazer as correções necessárias e devolve-lo o mais breve possível para o usuário fazer os acertos.

Na Figura 37, visualiza-se a interface “Registro de Ponto”, que possibilita consultar ou fazer a manutenção das ocorrências registradas e das situações apuradas pelo sistema.

Dia Semana	Data	Tipo Dia	Escala base	Hr.Ent.1	Hr.Saí.1	Hr.Ent.2	Hr.Saí.2
Quarta	03/09/03	Trabalhado	6	07:00	12:00	13:00	17:45
Quinta	04/09/03	Trabalhado	6	07:00	12:00	13:00	17:45
Sexta	05/09/03	Trabalhado	6	07:00	12:00		
[Sábado]	06/09/03	Compensado	6				
[Domingo]	07/09/03	DSR	6				
Segunda	08/09/03	Trabalhado	6	07:00	12:00	13:00	17:45
Terça	09/09/03	Trabalhado	6	07:00	12:00	13:00	17:45

Dt.Batida	Hr.Batida	Local	Relógio	Situação	Descrição situação	Origem	Qtd.Apurada	B.H.	Rateio	A/J
09/09/03	06:00	02094E	0	1	Trabalho diurno		10:45		02094E	
09/09/03	06:50	02094E	0	4	Extra Antes - Diurna		0:50		02094E	
09/09/03	07:00	02094E	0	19	Extra após - Diu		3:00		02381E	
09/09/03	18:00	02094E	0	20	Extra após - Not		1:00		02381E	
09/09/03	19:00	02381E	0							
09/09/03	23:00	02381E	0							

Figura 37 – Interface Registro de Ponto

Na parte superior da interface tem-se uma grade contendo as informações básicas dos registros diários de ponto correspondente ao período. Estas informações não podem ser alteradas, são atualizadas automaticamente durante a coleta e apuração do ponto.

Na parte inferior esquerda da interface há uma grade contendo as ocorrências de ponto do colaborador no dia selecionado, ou seja, os registros que o colaborador efetuou no relógio ponto eletrônico. A coluna “Local” demonstra o código do local em foi registrado ponto. A coluna “Relógio” demonstra o código do relógio em que foi registrado o ponto.

Na parte inferior direita da interface há uma grade contendo as situações apuradas automaticamente pelo sistema e/ou as situações incluídas pelo usuário. A coluna “Origem” refere-se a apuração. Se a situação foi apurada automaticamente pelo sistema a coluna conterà “ “ (vazio), se a situação foi incluída pelo usuário conterà “U”. Se a situação foi apurada pelo sistema mas o usuário fez alguma alteração será apresentada “A” de “Alterada”. A coluna “Qtd.Apurada” contém a quantidade de horas que o sistema calculou para a situação.

Na coluna “Rateio” é gravado o código do local para o qual as horas apuradas deverão ser consideradas. No exemplo apresentado na Figura 37 verifica-se que o colaborador registrou o ponto no local 02094E e 02381E.

Considerando-se que os colaboradores às vezes se ausentam do trabalho por motivos diversos, não registrando o ponto, faz-se necessário o cadastramento destes afastamentos. Para cadastrar o afastamento de um colaborador deve-se selecionar a opção “Cadastro...” no sub-menu “Afastamentos”.

Toma-se como exemplo a inclusão do afastamento conforme a Figura 38, onde o colaborador se ausentou do trabalho entre os dias 26/08/2003 e 06/09/2003 entregando ao departamento pessoal da empresa um atestado médico para justificar a falta.

The screenshot shows a software window titled "Afastamentos". At the top, there are three icons: a blue square with a white 'A', a green circle with a white checkmark, and a red circle with a white 'X'. Below the title bar, the text "Afastamentos" is displayed. Underneath, the employee information is shown: "Colaborador" followed by a text box containing "019682" and the name "MARCELO JOSÉ GONÇALVES". There are two tabs: "Consulta" and "Inclusão", with "Inclusão" being the active tab. The form contains several input fields: "Data/Hora início" with the value "26/08/2003" and "07:00"; "Data/Hora final" with the value "06/09/2003" and "23:59"; "Situação" with the value "37" and the text "Atestado"; and "Observações" with the text "ENTREGOU O ATESTADO EM 07/09/2003".

Figura 38 – Interface Afastamentos

Os atributos da interface para o cadastramento do afastamento são descritos a seguir:

- a) data/hora início: informar a data e hora de início da justificativa do afastamento do colaborador;
- b) data/hora final: informar a data e hora de término da justificativa do afastamento. Para algumas situações de afastamento não é necessário informa o término na inclusão. Por exemplo: se o colaborador estiver afastado em situação de auxílio doença, só informar a Data/Hora final quando o colaborador retornar do afastamento efetivamente;
- c) situação: informar o código da situação que justificará o afastamento do colaborador no período informado.

Para o cadastramento de afastamentos que se aplicam a todos os colaboradores da empresa, deve-se acessar a opção “Coletivo...” no sub-menu “Afastamentos”. Por exemplo, ao final do ano, quando a empresa concede férias coletivas a todos os colaboradores em um mesmo período.

Para encerrar as funcionalidades do sistema, a seguir é explicado como o sistema controla o banco de horas dos colaboradores.

A interface para manutenção do banco de horas dos colaboradores é acessada através da opção “Cadastro...” no sub-menu “Banco de Horas”.

A utilização do banco de horas faz-se necessária quando a empresa opta em não enviar para a folha de pagamento as horas extras apuradas no período corrente. Isso fará com que o sistema registre no banco de horas um “crédito” de horas quando apurar situações de horas

extras ou um “débito” quanto apurar faltas. O “débito” só é efetivado cadastrando-se um afastamento com situação 48-Débito banco de horas na interface “Afastamentos”.

### 3.4 RESULTADOS E DISCUSSÃO

Conforme as características dos sistemas consultados, não foram encontrados sistemas de apuração de ponto eletrônicos que utilize o banco de dados PostgreSQL, uma solução viável para empresas de qualquer porte, já que é software livre.

Também não foram encontrados sistemas que automatizem o procedimento de troca de escala de horário conforme o local de trabalho sem a intervenção do usuário do sistema. Nos sistemas pesquisados a troca de escala de horário do colaborador deve ser feita manualmente pelo usuário do sistema.

No sistema implementado foi possível automatizar esta tarefa através do registro de ponto dos colaboradores. O sistema ainda calcula a quantidade de horas proporcionalmente, para cada situação, em cada local onde foi registrado o ponto.

Considerando o registro de ponto testado conforme a Tabela 2, os resultados obtidos na apuração a partir do sistema podem ser verificados na Tabela 3, onde visualiza-se o rastreamento das atividades desempenhadas pelos colaboradores.

Tabela 2 – Registro de Ponto

<b>Empresa</b>	<b>Cadastro</b>	<b>Data</b>	<b>Hora</b>	<b>Relógio</b>	<b>Local</b>
0001	016624	26/08/2004	07:30	05	02094
0001	016624	26/08/2004	10:00	05	02094
0001	016624	26/08/2004	11:00	06	
0001	016624	26/08/2004	16:00	06	
0001	016624	26/08/2004	16:00	06	03089
0001	001624	26/08/2004	18:45	06	03089

Tabela 3 – Situações Apuradas e Rateadas

<b>Situação apurada</b>	<b>Qtd. Horas</b>	<b>Rateio/Local</b>
Trabalho diurno	2:30	02094
Trabalho diurno	4:00	
Trabalho diurno	1:45	03089
Atraso na entrada	0:30	02094
Extras saída	1:00	03089
Saída intermediária	0:30	02094
Saída intermediária	0:30	

Apesar de não ter sido proposto inicialmente o sistema ainda gera informações para controlar o banco de horas dos colaboradores. As horas extras apuradas pelo sistema podem ser armazenadas como crédito e mediante programação de afastamento feita pelo usuário, estas podem ser debitadas em benefício do colaborador.

## 4 CONCLUSÕES

É verdade que existem vários softwares para auxiliar no apontamento de frequência de ponto de colaboradores, mas os softwares testados pela empresa não satisfizeram as necessidades operacionais específicas da mesma, principalmente nos aspectos: troca de escala automática conforme local de trabalho, rateio conforme local de trabalho e utilização de SGBD livre.

O desenvolvimento deste trabalho possibilitou aperfeiçoar o conhecimento de diversos softwares como: a linguagem de programação *Object Pascal* e o ambiente Delphi 7, a biblioteca de acesso a banco de dados ZeosDBO, o sistema operacional Unix através do emulador Cygwin, a ferramenta de projeto banco de dados Power Designer 10 e o SGBD PostgreSQL.

O principal objetivo do trabalho, desenvolver um software de apuração de ponto eletrônico utilizando o banco de dados PostgreSQL foi alcançado. A manipulação da base de dados ficou muito fácil utilizando a biblioteca ZeosDBO.

Os objetivos específicos do trabalho, também foram alcançados, como o desenvolvimento do algoritmo para o tratamento das horas para calcular a quantidade de horas em 48 situações apresentadas no Apêndice E, o rateio das horas calculadas por local trabalhado, a troca automática de escala conforme o local de trabalho registrado no ponto.

O algoritmo para o cálculo de horas ficou um pouco extenso, mas é possível tratar as marcações de entrada e/ou saída independente da quantidade de vezes diária que os colaboradores efetuarem seu registro de ponto.

Acreditando na expansão e no crescimento dos softwares ditos softwares livres, este trabalho possibilitou desenvolver o sistema utilizando o banco de dados PostgreSQL, aproveitando assim as características e vantagens apresentadas, tais como: não ter custo para licenciamento, ser um SGBD estável e confiável e compatibilidade com vários sistemas operacionais.

### 4.1 EXTENSÕES

Como sugestão para trabalhos futuros o SI implementado poderia comunicar-se diretamente com os relógios eletrônicos ou coletores de dados eliminando assim o processo de

coleta do registro de ponto. Assim cada vez que um colaborador registrar o ponto, o sistema poderia imediatamente fazer a apuração das horas e também disponibilizar no local de trabalho o saldo de horas que o colaborador possui no período.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADVOCACY. **As vantagens do PostgreSQL**, [s.l.], 2002. Disponível em: <<http://advocacy.postgresql.org/advantages/?lang=br>>. Acesso em: 03 nov 2004.
- CYGWIN. **What versions of Windows are supported?**, [s.l.], 2004a. Disponível em: <[http://cygwin.com/faq/faq\\_2.html#SEC6](http://cygwin.com/faq/faq_2.html#SEC6)>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **What are the licensing terms?**, [s.l.], 2004b. Disponível em: <[http://cygwin.com/faq/faq\\_2.html#SEC6](http://cygwin.com/faq/faq_2.html#SEC6)>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **Installation instructions**, [s.l.], 2004c. Disponível em: <[http://cygwin.com/faq/faq\\_2.html#SEC6](http://cygwin.com/faq/faq_2.html#SEC6)>. Acesso em: 03 nov 2004.
- DATE, C.J. **Introdução a sistemas de bancos de dados**. Tradução Contexto Traduções Ltda. Rio de Janeiro: Campus, 1991.
- DBEXPERTS. **Características do dbExperts PostgreSQL 7.3.2**, [s.l.], 2004. Disponível em: <<http://www.dbexperts.com.br/postgresql/professional/caract>>. Acesso em: 03 nov 2004.
- DEFAULT. **Características SIPE**, [s.l.], 2004. Disponível em: <<http://www.default.com.br/pcharacteristicas.html>>. Acesso em: 03 nov 2004.
- DIGILOGO. **Tecnoponto**: ponto eletrônico por computador, [s.l.], 2004. Disponível em: <[http://www.tecnoponto.hpg.ig.com.br/pg\\_1.htm](http://www.tecnoponto.hpg.ig.com.br/pg_1.htm)>. Acesso em: 03 nov 2004.
- FSF - FREE SOFTWARE FOUNDATION. **GNU General Public Licence**, [s.l.], 2004a. Disponível em: <<http://www.fsf.org/licenses/gpl.html>>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **O manifesto GNU de Richard Stallman**, [s.l.], 2002a. Disponível em <<http://www.fsf.org/gnu/initial-announcement.pt.html>>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **Why you shouldn't use the library GPL for your next library**, [s.l.], 2004b. Disponível em: <<http://www.fsf.org/licenses/why-not-lgpl.html>>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **Introdução ao projeto GNU**, [s.l.], 2002b. Disponível em: <<http://www.fsf.org/gnu/gnu-history.pt.html>>. Acesso em: 03 nov 2004.
- \_\_\_\_\_. **O que é software livre?**, [s.l.], 2003. Disponível em: <<http://www.fsf.org/philosophy/free-sw.pt.html>>. Acesso em: 03 nov 2004.
- LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informações gerenciais: administrando a empresa digital**. Tradução Arlete Simille Marques. São Paulo: Prentice Hall, 2004.

LCS. **WinPto v7.0**, [s.l.], 2004. Disponível em:

<<http://www.lcsinformatica.com.br/produtos/winpto.htm>>. Acesso em: 03 nov 2004.

MOMJIAN, Bruce. **PostgreSQL: introduction and concepts**. Boston: Addison Wesley Longman, 2001.

NASCIMENTO, Amauri Mascaro. **Iniciação ao direito do trabalho**. 30. ed. São Paulo: LTr, 2004.

NEVES, Denise Lemes Fernandes. **PostgreSQL: conceitos e aplicações**. São Paulo: Érica, 2002.

NORBER. **NewPonto**, [s.l.], 2004. Disponível em:

<<http://www.norber.com.br/pages/Produto.html>>. Acesso em: 03 nov 2004.

OSI - OPEN SOURCE INITIATIVE. **Basic idea behind open source**, [s.l.], 2004a.

Disponível em: <<http://www.opensource.org/index.php>>. Acesso em: 03 nov 2004.

\_\_\_\_\_. **The approved licenses**, [s.l.], 2004b. Disponível em:

<<http://www.opensource.org/licenses/>>. Acesso em: 03 nov 2004.

POSTGRESQL. **PostgreSQL is released under the BSD licence**, [s.l.], 2004. Disponível em: <<http://www.postgresql.org/licence.html>>. Acesso em: 03 nov 2004.

REZENDE, Denis Alcides. **Planejamento de sistemas de informação e informática: guia prático para planejar a tecnologia da informação integrada ao planejamento estratégico das organizações**. São Paulo: Atlas, 2003.

ROSINI, Alessandro Marco; PALMISANO, Ângelo. **Administração de sistemas de informação e a gestão do conhecimento**. São Paulo: Thomson, 2003.

STAIR, Ralph M.; REYNOLDS, Georg W. **Princípios de sistemas de informação: uma abordagem gerencial**. Tradução Alexandre Melo de Oliveira. 4. ed. Rio de Janeiro: LTC, 2002.

TASK. **Soluções de frequência**, [s.l.], 2004. Disponível em:

<[http://www.tasksistemas.com.br/ta\\_solucao.php?id\\_solucao=1](http://www.tasksistemas.com.br/ta_solucao.php?id_solucao=1)>. Acesso em: 03 nov 2004.

WORSLEY, John C; DRAKE, Joshua D. **Practical PostgreSQL**. Sebastopol, CA: O'Reilly, 2002.

ZEOS Library. **Open-source tools for you database solutions**, [s.l.], 2003. Disponível em: <<http://www.zeoslib.net/modules.php?name=Content&pa=showpage&pid=5>>. Acesso em 03 nov 2004.

## APÊNDICE A – Dicionário de Dados.

PK – chave primária, FK – chave estrangeira

<b>Tabela AFASTAMENTO: programação de afastamento e débito banco de horas feita pelo usuário do sistema para cada colaborador.</b>				
CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
DT_INI_AFA	DATE	not null	PK	Data de início em que o colaborador irá se afastar
HR_INI_AFA	TIME	not null	PK	Hora de início em que o colaborador irá se afastar
CD_COD_SIT	INT4	not null	FK	Código da Situação
DT_FIM_AFA	DATE	null		ata de término do afastamento
HR_FIM_AFA	TIME	null		Hora de término
DS_OBS_AFA	VARCHAR(40)	null		Breve descrição referente ao afastamento

<b>Tabela BANCODEHORA: armazena registro referente ao banco de horas dos colaboradores. Contém o histórico dos lançamentos que creditaram/debitaram em BCOHORAS.</b>				
CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_COD_CAL	INT4	not null	PK,FK	Código do período de Cálculo
CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
DT_MAR_BHC	DATE	not null	PK	Data da ocorrência do débito/crédito do Banco de Horas
FG_ORI_BHC	VARCHAR(1)	not null	PK	Pode conter vazio, "D" ou "A"
CD_COD_LOC	VARCHAR(6)	null	FK	Código do Local
CD_COD_REL	INT4	null	FK	Código do coletor/relógio
CD_COD_SIT	INT4	null	FK	Código da Situação
QT_HRS_BHC	VARCHAR(6)	not null		Quantidade de horas apuradas pela situação que gerou a ocorrência no Banco de Horas
FG_B_H_BHC	VARCHAR(1)	not null		Flag que indica se a ocorrência do Banco de Horas deve ser Debitada ou Creditada
DS_OBS_BHC	VARCHAR(40)	null		Breve descrição referente ao registro da Ocorrência

<b>Tabela CALCULO: armazena Períodos de Calculo para apuração e período de calculo para integração com sistema de folha de pagamento</b>				
CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_COD_CAL	INT4	not null	PK	Código do período de Cálculo
DT_INI_CAL	DATE	not null		Data inicial do período de apuração
DT_FIM_CAL	DATE	not null		Data final do período de apuração
DT_IFP_CAL	DATE	not null		Data inicial do período de calculo de folha de pagamento
DT_FFP_CAL	DATE	not null		Data final do período de calculo de folha de pagamento
FG_APU_CAL	VARCHAR(1)	not null		Indica a situação da apuração
FG_BHR_CAL	VARCHAR(1)	not null		Indica se as ocorrências do

				Banco de horas já foram processadas
FG_PER_CAL	VARCHAR(1)	not null		Situação do calculo em relação à exportação dos eventos folha

**Tabela CIDADE: armazena registros referente cidades**

CD_COD_CID	INT4	not null	PK	Código da cidade
DS_NOM_CID	VARCHAR(40)	not null		Nome da cidade
DS_U_F_CID	VARCHAR(2)	not null		Sigla da UF da cidade
CD_NAC_CID	INT4	null		Código nacional

**Tabela COLABORADOR: armazena os colaboradores que têm seu registro de ponto controlado pelo sistema**

CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
CD_COD_FNC	INT4	not null	FK	
CD_COD_ESC	INT4	not null	FK	Código da escala
CD_COD_SEM	INT4	not null	FK	
DS_NOM_COL	VARCHAR(40)	not null		Nome do colaborador
DT_ADM_COL	DATE	not null		Data de admissão do colaborador
FG_SAL_COL	VARCHAR(1)	not null		Indica se o colaborador é Mensalista ou Horista
FG_B_H_COL	VARCHAR(1)	not null		Indica se o colaborador é optante pelo Banco de Horas
FG_ESC_COL	VARCHAR(1)	not null		Indica se a escala de horário do colaborador é fixa ou não
DT_CAD_COL	DATE	not null		Data cadastro do colaborador
CD_SFP_COL	INT4	null		Código de situação de afastamento do colaborador no sistema de folha de pagamento
DT_SFP_COL	DATE	null		Data em que ocorreu a situação de afastamento atual

**Tabela DATASESCALA: horário definido para uma data em específico**

CD_COD_ESC	INT4	not null	PK,FK	Código da escala
DT_DEF_ESC	DATE	not null	PK	Data Definida da Escala
HR_EN1_ESC	TIME	not null		Hora de Entrada da 1º parte
HR_SA1_ESC	TIME	not null		Hora de Saída da 1º parte
HR_EN2_ESC	TIME	null		Hora de Entrada da 2º parte
HR_SA2_ESC	TIME	null		Hora de Saída da 2º parte

**Tabela DATASFERIADO: armazena as datas da tabela "pai" feriado**

CD_COD_FER	INT4	not null	PK,FK	Código da tabela de feriado
CD_ANO_FER	INT4	not null	PK	Ano corresponde ao feriado
DT_DIA_FER	DATE	not null	PK	Data do feriado
DS_NOM_FER	VARCHAR(20)	not null		Descrição do Feriado

**Tabela EMPRESA: armazena o cadastro de empresas do sistema**

CD_COD_EMP	INT4	not null	PK	Código da Empresa
CD_COD_CID	INT4	not null	FK	Código da cidade
DS_RAZ_EMP	VARCHAR(40)	not null		Nome/Razão Social da Empresa
CD_CGC_EMP	INT8	null		CNPJ da empresa

CD_INS_EMP	INT8	null		Inscrição Estadual da Empresa
DS_END_EMP	VARCHAR(40)	not null		Descrição do Endereço
DS_BAI_EMP	VARCHAR(20)	not null		Descrição do bairro
CD_CEP_EMP	INT4	not null		CEP da empresa
CD_JUS_EMP	INT4	null		Código do Evento de Horas Justificadas
CD_ABO_EMP	INT4	null		Código do evento de Horas Abonadas
CD_HNP_EMP	INT4	null		Código do Evento de Horas Normais Projetadas
CD_DSP_EMP	INT4	null		Código dos Eventos de DSR e Feriado Projetado
FG_PRJ_EMP	VARCHAR(1)	null		Indica se é para projetar as horas do período de calculo
FG_B_H_EMP	VARCHAR(1)	null		Indica se o banco de horas será utilizado pela empresa

**Tabela ESCALA: armazena as escalas de horários**

CD_COD_ESC	INT4	not null	PK	Código da escala
CD_COD_FER	INT4	not null	FK	Código do feriado
DS_NOM_ESC	VARCHAR(30)	not null		Descrição da escala de horário
DS_OBS_ESC	VARCHAR(30)	null		Descrição de observações
QT_DSR_ESC	NUMERIC(5,2)	not null		Quantidade de horas DSR
QT_SEM_ESC	NUMERIC(5,2)	not null		Quantidade de horas semanais
HR_NTE_ESC	TIME	not null		Hora de início horário noturno
HR_NTS_ESC	TIME	not null		Hora de termino horário noturno
HT_FE1_ESC	TIME	null		Qtdade tolerância para Falta na Entrada da 1ª parte do horário
HT_FS1_ESC	TIME	null		Qtdade tolerância para Falta na Saída da 1ª parte do horário
HT_FE2_ESC	TIME	null		Qtdade tolerância para Falta na Entrada da 2ª parte do horário
HT_FS2_ESC	TIME	null		Qtdade tolerância para Falta na Saída da 2ª parte do horário
HC_EE1_ESC	TIME	null		Qtdade para considerar Extra na Entrada da 1ª parte do horário
HC_ES1_ESC	TIME	null		Qtdade para considerar Extra na Saída da 1ª parte do horário
HC_EE2_ESC	TIME	null		Qtdade para considerar Extra na Entrada da 2ª parte do horário
HC_ES2_ESC	TIME	null		Qtdade para considerar Extra na Saída da 2ª parte do horário
TP_SEG_ESC	VARCHAR(1)	not null		Tipo Segunda: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_2E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_2S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_2E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_2S2_ESC	TIME	null		Hora de Saída da 2ª parte

TP_TER_ESC	VARCHAR(1)	not null		Tipo Terça: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_3E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_3S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_3E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_3S2_ESC	TIME	null		Hora de Saída da 2ª parte
TP_QUA_ESC	VARCHAR(1)	not null		Tipo Quarta: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_4E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_4S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_4E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_4S2_ESC	TIME	null		Hora de Saída da 2ª parte
TP QUI_ESC	VARCHAR(1)	not null		Tipo Quinta: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_5E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_5S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_5E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_5S2_ESC	TIME	null		Hora de Saída da 2ª parte
TP_SEX_ESC	VARCHAR(1)	not null		Tipo Sexta: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_6E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_6S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_6E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_6S2_ESC	TIME	null		Hora de Saída da 2ª parte
TP_SAB_ESC	VARCHAR(1)	not null		Tipo Sábado: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_7E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_7S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_7E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_7S2_ESC	TIME	null		Hora de Saída da 2ª parte
TP_DOM_ESC	VARCHAR(1)	not null		Tipo Domingo: [T]rabalho, [N]ormal, [F]olga, [C]ompensado, [D]-DSR
HR_1E1_ESC	TIME	null		Hora de Entrada da 1ª parte
HR_1S1_ESC	TIME	null		Hora de Saída da 1ª parte
HR_1E2_ESC	TIME	null		Hora de Entrada da 2ª parte
HR_1S2_ESC	TIME	null		Hora de Saída da 2ª parte

**Tabela FERIADO: tabela "PAI" do cadastro de Feriados**

CD_COD_FER	INT4	not null	PK	Código tabela de feriado
DS_TAB_FER	VARCHAR(20)	not null		Descrição da tabela de feriado

**Tabela FUNCAO: armazena registro de funções/cargos**

CD_COD_FNC	INT4	not null	PK	
DS_NOM_FNC	VARCHAR(40)	null		

**Tabela LOCAL: registro de parametrização de local**

CD_COD_LOC	VARCHAR(6)	not null	PK	Código do Local
CD_COD_ESC	INT4	null	FK	Código da escala
CD_COD_FER	INT4	null	FK	Código do Feriado
DS_NOM_LOC	VARCHAR(25)	not null		Descrição do Local
CD_RAT_LOC	INT4	null		Código do Rateio

HR_NTE_LOC	TIME	null		Hora Entrada horário noturno
HR_NTS_LOC	TIME	null		Hora Saída horário noturno
DT_ATU_LOC	DATE	not null		Data de atualização

**Tabela MARCACAO: registro "PAI" das ocorrências de ponto em cada dia**

CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_COD_CAL	INT4	not null	PK,FK	Código do período de cálculo
CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
DT_DIA_MAR	DATE	not null	PK	Data que se referem às ocorrências de ponto
CD_COD_ESC	INT4	null	FK	Código da escala
TP_DIA_MAR	VARCHAR(1) null			Tipo Dia: [f]olga, [C]ompensado, [D]SR, [T]rabalhado, [F]eriado e Vazio (Não apurado)
HR_EN1_MAR	TIME	null		Hora entrada usada na apuração
HR_SA1_MAR	TIME	null		Hora saída usada na apuração
HR_EN2_MAR	TIME	null		Hora entrada usada na apuração
HR_SA2_MAR	TIME	null		Hora saída usada na apuração

**Tabela MAROCORRENCIA: registro de ocorrências do ponto (filha de marcação) dos colaboradores.**

CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
CD_COD_CAL	INT4	not null	PK,FK	Código do período de Cálculo
DT_DIA_MAR	DATE	not null	PK,FK	Data que se referem às ocorrências
CD_SEQ_MAR	INT4	not null	PK	Código sequencial
CD_COD_ESC	INT4	null	FK	Código da escala
DT_BAT_MAR	DATE	null		Data de registro de ponto
HR_BAT_MAR	TIME	null		Hora de registro de ponto
CD_LOC_MAR	VARCHAR(6)	null		Local que foi registrado ponto
CD_REL_MAR	INT4	null		Relógio que registrou o ponto

**Tabela MARORIGINAL: armazena ocorrências originais coletadas/importada do(s) arquivo(s) texto dos relógios ponto e/ou coletores**

CD_SEQ_ORI	INT4	not null	PK	Código sequencial
CD_COD_REL	INT4	null	FK	Código do coletor/relógio
CD_COD_EMP	INT4	null	FK	Código da Empresa
CD_CAD_COL	INT4	null	FK	Código do colaborador
CD_COD_LOC	VARCHAR(6)	null	FK	Código do Local
DT_MAR_ORI	DATE	null		Data de registro de ponto
HR_MAR_ORI	TIME	null		Hora de registro de ponto

**Tabela RELOGIO: definição da escala de horário para Relógio/Coletores**

CD_COD_REL	INT4	not null	PK	Código do coletor/relógio
CD_COD_ESC	INT4	null	FK	Código da escala
CD_COD_FER	INT4	null	FK	Código do Feriado
DS_NOM_REL	VARCHAR(25)	not null		Descrição do coletor/relógio
HR_NTE_REL	TIME	null		Hora Entrada do horário noturno
HR_NTS_REL	TIME	null		Hora Saída do horário noturno
DT_ATU_REL	DATE	not null		Data de atualização

**Tabela SITAPURADA: situações apuradas no dia. "Filha" de marcação**

CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_COD_CAL	INT4	not null	PK,FK	Código do período de Cálculo

CD_CAD_COL	INT4	not null	PK,FK	Código do colaborador
DT_DIA_MAR	DATE	not null	PK,FK	Data que se referem às ocorrências
CD_SEQ_APU	INT4	not null	PK	Código sequencial
CD_COD_SIT	INT4	not null	FK	Código da Situação
FG_ORI_APU	VARCHAR(1)	null		Origem da apuração
QT_HRS_APU	VARCHAR(6)	null		Quantidade de horas apuradas
CD_LOC_APU	VARCHAR(6)	null		Local para rateio
FG_B_H_APU	VARCHAR(1)	null		Flag Banco de Horas
FG_ABO_APU	VARCHAR(1)	null		Flag [A]bonado/[J]ustificado

**Tabela SITUACAO: Tabela de parametrização das situações para apuração/tratamento de horas**

CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
CD_COD_SIT	INT4	not null	PK	Código da Situação
DS_NOM_SIT	VARCHAR(30)	not null		Descrição da Situação
FG_DSR_SIT	VARCHAR(1)	null		Flag Perde DSR (S/N)
FG_FER_SIT	VARCHAR(1)	null		Flag Perde Férias (S/N)
FG_ABO_SIT	VARCHAR(1)	null		Flag Justifica/Abona (S/N)
FG_EXC_SIT	VARCHAR(1)	null		Flag Exceção (S/N)
FG_APU_SIT	VARCHAR(1)	null		Flag Apurar (S/N)
FG_BHC_SIT	VARCHAR(1)	null		Flag Banco de Horas (+/-)
TX_BHC_SIT	NUMERIC(4,2)	null		Multiplicador Banco de Horas
CD_EBH_SIT	INT4	null		Código Evento Banco de Horas
CD_EVE_SIT	INT4	null		Código Evento período atual
CD_EVA_SIT	INT4	null		Código Evento período anterior

**Tabela SUBEMPRESA: registros de sub-empresas das empresas do sistema**

CD_COD_SEM	INT4	not null	PK	Código da SubEmpresa
CD_COD_EMP	INT4	not null	PK,FK	Código da Empresa
DS_RAZ_SEM	VARCHAR(40)	not null		Descrição do nome da SubEmpresa

## APÊNDICE B – Passos para instalação do Cygwin no Windows.

O download do programa de instalação do “Cygwin Setup” pode ser obtido na Internet no link <http://cygwin.com/setup.exe>.

Conforme instruções do programa “Cygwin Setup” os passos necessários para a instalação do Cygwin são descritos abaixo:

- 1) Execute o arquivo “setup.exe” e na janela intitulada “Cygwin Net Release Setup Program” clique no botão avançar;
- 2) Na janela seguinte “Choose A Download Source” é solicitado a localização dos arquivos de instalação. Escolha a opção “install from Internet” e clique avançar;
- 3) Na próxima janela “Select Root Install Directory” informe o diretório root como C:\cygwin e clique avançar;
- 4) Na janela “Select Local Package Directory” informe uma pasta onde serão armazenados os arquivos que serão baixados da Internet, ex: C:\temp, clique avançar;
- 5) Na janela “Select Your Internet Connection” selecione “Direct Connection” e clique avançar;
- 6) Na janela “Choose A Download Site” selecione a localização da Internet de onde serão baixados os arquivos de instalação e clique em avançar;
- 7) Na janela seguinte “Select Packages” devem ser selecionados os pacotes a serem instalados. Clique no botão *view* até que apareça “Full”. Localize o na coluna *Package* “*postgresql: PostgreSQL Data Base Management System*” e clique na coluna *New* até que apareça a versão do SGBD a ser instalado. Clique no botão avançar;
- 8) Na janela denominada “Create Icon” deixe a opção “Create a icon on Desktop” selecionada para que seja criado um atalho na sua área de trabalho e clique no botão concluir;
- 9) Uma ultima janela “Installation Complete” será apresentada informando que a instalação do emulador Cygwin foi concluída.

Abaixo são descritos os passo para a instalação do Cygipc que pode ser obtido na internet no endereço <<http://www.neuro.gatech.edu/users/cwilson/cygutils/cygipc/cygipc-2.00.tar.bz2>>.

Faça download do arquivo e salve na pasta de instalação do Cygwin. Abra um shell do Cygwin e descompacte conforme abaixo:

- 1) Vá para a pasta raiz do ambiente Cygwin digitando: `cd / [ENTER]`;
- 2) Descompacte o arquivo da seguinte forma: `tar xvfj cygipc-2.00.tar.bz2 [ENTER]`.

## APÊNDICE C – Método pInsertCommit da classe TfrEscolhe.

```

procedure TfrEscolhe.pInsertCommit(prNrTabela : integer);
var
  wCamposSql, wValuesSql : string;
  wComponente : TComponent;
  i,j : integer;
  wNrCampoChave : integer;

begin
  //limpa instrução sql do componente ZeosDBO (TZQuery)
  queryManutencao.SQL.Clear;
  //insere primeira linha no componente sql
  queryManutencao.SQL.Add('INSERT INTO ' + wNmTabela + ' ');

  //monta o nome dos campos do comando insert
  wCamposSql:='(';
  //monta o conteudo dos campos
  wValuesSql:='VALUES (';
  //numero da tabela no sistema
  i:=prNrTabela;

  //percorre segunda dimensao do array
  for j := 0 to Length(wTabelas[i])-1 do
    begin
      //Verifica se o campo pertence a tabela
      if UpperCase(wTabelas[i,j].NomeTabela) = UpperCase(wNmTabela) then
        begin
          //se existir um componente de edicao no form correspondente ao campo
          wComponente:=fPegaComponente(wTabelas[i,j].NomeCampo);
          if wComponente <> Nil then
            begin
              //coloca virgula apos cada campo concatenado
              if wCamposSql <> '(' then
                begin
                  wCamposSql := wCamposSql + ',';
                  wValuesSql := wValuesSql + ',';
                end;
              //acrescenta o nome do campo no sql
              wCamposSql := wCamposSql + wTabelas[i,j].NomeCampo;

              //se é para gerar codigo automaticamente
              if wGeraCodigoAutomatico then
                begin
                  //acrescente o conteudo dos campos no sql (em VALUES(...))
                  wNrCampoChave := fIDCampoChavePri(wTabelas[i,j].NomeCampo);
                end;
            end;
        end;
    end;
  end;

```

```

//se o campo for correspondente ao último campo da chave primária
if ( wNrCampoChave = Length(wChavePri)-1 ) and
  //e se o campo for tipo inteiro
  ( wTabelas[i,j].TipoCampo = 'I' ) then
begin
  //acrescenta o conteúdo do campo no sql
  wValuesSql := wValuesSql + wValorChavePri[wNrCampoChave];
end
else
//se nao for campo da chave primaria
begin
  //acrescenta o conteudo do campo no sql
  wValuesSql := wValuesSql + fPegaValorSQL(i, j, fPegaConteudoComponente(wComponente));
end;
end
else
//se nao é para gerar código automaticamente
begin
  //acrescenta o conteudo do campo no sql
  wValuesSql := wValuesSql + fPegaValorSQL(i, j, fPegaConteudoComponente(wComponente));
end;
end;
end;
end;

//finaliza campos e values
wCamposSql:=wCamposSql + ')';
wValuesSql:=wValuesSql + ')';

//adiciona trecho sql no componente ZeosDBO com os campos
queryManutencao.SQL.Add( wCamposSql );
//adiciona trecho sql no componente ZeosDBO com os valores
queryManutencao.SQL.Add( wValuesSql );

//executa a instrução sql através do componente ZeosDBO
queryManutencao.ExecSql;

end;

```

## APÊNDICE D – Trecho do método pFillFields da classe TfrEscolhe.

```

procedure TfrEscolhe.pFillFields(prNrTabela : integer);
var
  wComponente : TComponent;
  i, j : integer;
begin
  i:=prNrTabela;
  //percorre segunda dimensao do array
  for j := 0 to Length(wTabelas[i])-1 do
    begin
      //Verifica se o campo pertence a tabela
      if UpperCase(wTabelas[i,j].NomeTabela) = UpperCase(wNmTabela) then
        begin
          //se existir um componente de edicao na tela correspondente ao campo
          wComponente:=fPegaComponente(wTabelas[i,j].NomeCampo);
          if wComponente <> Nil then
            begin
              //pegar o conteudo dos campos e jogar nos objetos de edicao
              //se o objeto for Edit
              if UpperCase(wComponente.ClassName) = UpperCase('TLogiEdit') then
                TLogiEdit(wComponente).Text:=
                  queryManutencao.FieldByName(wTabelas[i,j].NomeCampo).asString
              else
                //se o objeto de edicao for TEditNumeroInteiro
                if UpperCase(wComponente.ClassName) = UpperCase('TLogiEditNumeroInteiro') then
                  TLogiEditNumeroInteiro(wComponente).SetValue(
                    queryManutencao.FieldByName(wTabelas[i,j].NomeCampo).asInteger)
                else
                  //se o objeto de edicao for TLogiEditFloat
                  if UpperCase(wComponente.ClassName) = UpperCase('TLogiEditFloat') then
                    TLogiEditFloat(wComponente).SetValue(
                      queryManutencao.FieldByName(wTabelas[i,j].NomeCampo).asFloat )
                  else
                    //se o objeto de edicao for tipo TEditEstado
                    if UpperCase(wComponente.ClassName) = UpperCase('TLogiEditEstado') then
                      TLogiEditEstado(wComponente).Text :=
                        queryManutencao.FieldByName(wTabelas[i,j].NomeCampo).asString
                    else
                      .....
                end;
            end;
          end;
        end;
      end;
    end;
  end;

```

## **APÊNDICE E – Relação das situações tratadas na apuração.**

Situações geradas no computo das horas calculadas na apuração:

- 01 - Trabalho Diurno
- 02 - Trabalho Noturno
- 03 - Extra Antes Noturna
- 04 - Extra Antes
- 05 - Extra Entrada 1o.
- 06 - Extra Entrada 1o. Noturna
- 07 - Atraso Entrada 1o.
- 08 - Saída Antecipa 1o.
- 09 - Extra Saída 1o.
- 10 - Extra Saída 1o. Noturna
- 11 - Falta 1o. período
- 12 - Falta 2o. período
- 13 - Extra Entrada 2o.
- 14 - Extra Entrada 2o. Noturna
- 15 - Atraso Entrada 2o.
- 16 - Saída Antecipa 2o.
- 17 - Extra Saída 2o.
- 18 - Extra Saída 2o. Noturna
- 19 - Extra Após
- 20 - Extra Após Noturna
- 21 - Extra em Compensado
- 22 - Extra em Compensado Noturna
- 23 - Extra em DSR
- 24 - Extra em DSR Noturna
- 25 - Extra em Feriado
- 26 - Extra em Feriado Noturna
- 27 - Extra em Folga Noturna
- 28 - Extra em Folga
- 29 - Compensado
- 30 - DSR
- 31 - Feriado
- 32 - Saída Intermediária
- 33 - Refeição
- 34 - Refeição Noturna
- 35 - Ocorrências ímpar
- 36 - Fora de Horário

Situações utilizadas para cadastrar o afastamento dos colaboradores:

- 37 - Atestado
- 38 - Auxilio doença
- 39 - Acidente trabalho
- 40 - Férias
- 41 - Férias coletivas
- 42 - Licença maternidade
- 43 - Licença paternidade
- 44 - Licença remunerada
- 45 - Licença sem remuneração
- 46 - Aviso prévio trabalhado
- 47 - Serviço militar
- 48 - Débito banco de horas