

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**SISTEMA DE BANCO DE DADOS PARA O  
GERENCIAMENTO DE SERVIÇOS EM  
CONCESSIONÁRIAS DE DISTRIBUIÇÃO DE  
ENERGIA ELÉTRICA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**Rodrigo Follmer**

**Santa Maria, RS, Brasil**

**2015**

**SISTEMA DE BANCO DE DADOS PARA O  
GERENCIAMENTO DE SERVIÇOS EM CONCESSIONÁRIAS  
DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA**

**POR**

**Rodrigo Follmer**

Trabalho de conclusão de curso de graduação apresentado ao Centro de Tecnologia da Universidade Federal de Santa Maria, como requisito parcial para obtenção do grau de **Bacharel em Engenharia de Produção**.

**Orientador: Profº. Dr. Vinicius Jacques Garcia**

**Santa Maria, RS, Brasil**

**2015**

# SISTEMA DE BANCO DE DADOS PARA O GERENCIAMENTO DE SERVIÇOS EM CONCESSIONÁRIAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

**Rodrigo Follmer (UFSM)**

rodrigofollmer@hotmail.com

**PROFº. DR. VINICIUS JACQUES GARCIA (UFSM)**

viniciusjg@ufsm.br

*Este trabalho visa o desenvolvimento de um banco de dados relacional para a gestão dos serviços de atendimento em uma concessionária de distribuição de energia elétrica. Os dados históricos obtidos serão na forma de um banco de dados relacional para possibilitar o acesso aos dados por meio de consultas em linguagem SQL. Dessa forma, a atribuição de serviços pendentes às equipes disponíveis torna-se mais facilitada em função da estruturação dos dados, que possibilitará o uso de uma interface de apoio ao usuário mais intuitiva e completa quanto à representação e à busca de informação relacionada aos serviços pendentes e aos recursos disponíveis.*

*Palavras-chave:* Banco de Dados; Distribuição de energia elétrica; Gestão de serviços; SQL; Sistema de suporte à decisão

*This work aims to develop a relational database for service management in electric utilities. The idea is to struct historical data in a database system in such a way that sql queries will be possible. With this approach, the assignment of a service to a given repair crew available becomes easier by adopting the data structure involved in the database developed. In addition, a future development of a user interface even more intuitive to handle this assignment problem may be suitable.*

*Keywords:* Databases; Electric power distribution; Service management; SQL; Decision-support system

## **1. Introdução**

A designação de recursos a demandas formam cenários típicos de gerenciamento de serviços. Relacionar corretamente capacidade com demanda envolve uma solução de despacho que tem origem nos centros de atendimento. Para tais soluções, espera-se que todas as demandas requisitadas para o sistema sejam atendidas. O problema ocorre, de forma análoga, em serviços de atendimento em concessionárias de energia elétrica. Nessas redes, existem centros de operação destinados ao atendimento de certas áreas de uma região, que recebem demandas (ordens de serviço) e contam com equipes (recursos internos) para atender às respectivas demandas. Cada alocação das ordens de serviço para uma equipe é denominada despacho, nesse sistema. Espera-se, através de um bom gerenciamento do serviço, que o centro de operações atenda a todas as demandas utilizando somente seus recursos internos.

Na prática, as ordens de serviço podem ser de diferentes naturezas. As tratadas neste trabalho são ordens comerciais e técnicas e ordens emergenciais. As ordens comerciais e técnicas possuem um prazo fixo para o seu atendimento, já as ordens emergenciais devem ser atendidas no menor prazo possível, possuindo prioridade maior em relação às outras ordens. Ou seja, ordens emergenciais se sobrepõem às ordens comerciais no que se diz respeito à prioridade de atendimento. Cabe aos gestores do serviço, portanto, a adaptação dessas ordens emergenciais nos despachos diários de cada equipe. As demandas emergenciais, portanto, geram mudanças nas rotas pré-programadas das equipes no dia. Esses atendimentos não previstos necessitam de critérios especiais para serem tratados de forma correta pelos centros e exigem uma atribuição e uma tomada de decisão por parte do centro de operações.

Sugere-se, nesse trabalho, meios de representar de uma forma estruturada os dados históricos obtidos de uma concessionária de energia elétrica da região central do Rio Grande do Sul. O sistema, modelado em um banco de dados relacional, procura desdobrar dados práticos (conjunto de informações não organizadas) que serão processados e transformados em base de apoio para a tomada de decisões. Será possível, também, sistematizar resultados de rotas obtidos para uma certa quantidade de dados de demanda e os recursos disponíveis no mesmo período. Ainda, o banco de dados permite armazenar essas soluções de despacho em uma interface de acesso. Através de consultas ao banco são obtidas informações que fornecem apoio à gestão do despacho de serviços em concessionárias. Isso representa, na literatura, um sistema de suporte à decisão para gerenciamento de serviços, tema bastante estudado e com vasta bibliografia.

Não é incomum, no contexto atual, referências na literatura sobre a importância da gestão de dados e como, se desprezada, ela pode dificultar o processo decisório. Essas referências serão apresentadas no próximo capítulo. A tarefa de capturar, sintetizar e transformar dados em informações complica ao passo que tais dados ganham volume e surgem necessidades como alocação de demandas emergenciais sob pressão de tempo e *stress*.

Não é do interesse do presente trabalho focar em questões complexas de programação e sistemas de banco de dados em si, mas sim exemplificar como eles podem se tornar eficazes no sentido de organização de dados históricos e suas peculiaridades relacionais.

O trabalho se justifica no momento em que o apoio à gestão dos despachos de serviços e a representação dos dados nos centros de operação precisa ser estruturado de forma a facilitar a consulta de designação das demandas com os recursos. Uma justificativa prática seria a de agregar todas essas informações dentro de uma estrutura de banco de dados relacional de modo que possam ser capturadas, vinculadas e interpretadas.

O objetivo aqui é, portanto, o desenvolvimento, projeto e organização dos dados dos serviços de atendimento na forma de um banco de dados para representar o conjunto de dados, de forma que as informações do serviço estejam adequadas à realidade das concessionárias.

Ao fim do trabalho, se busca a interpretação dos dados vinculados na forma de um banco de dados relacional para base de dados semelhantes. Para atingir o resultado, serão seguidas algumas etapas metodológicas. Inicialmente, Buscou-se estudos do assunto na literatura, estudo da natureza do problema e a definição do modelo de dados a ser adotado. Ainda, a construção do banco de dados e o estudo de caso modelando os dados reais obtidos.

## **2. Referencial teórico**

### **2.1. Bancos de dados: programação, funcionamento, armazenamento e gestão**

Com mercados competitivos de distribuição de energia elétrica emergindo pelo mundo, se torna necessário o desenvolvimento de soluções efetivas (sistemas ou algoritmos) para a gestão de recursos e serviços nesse setor (BERGEY, RAGSDALE E HOSKOTE, 2003). Ainda de acordo com os autores, a onipresença de Pacotes Office, combinando-se com o poder de sua linguagem de programação e gestão de dados, gera uma grande plataforma para a criação de sistemas de suporte à decisão. Mendonza, Medaglia e Velasco (2009) também relatam que o uso de aplicativos comerciais, como o MS Excel e Access podem ganhar

espaço nos serviços de atendimento das concessionárias, em detrimento de módulos customizados e algoritmos desenvolvidos para cada empresa, o que envolve alto custo e pouca flexibilidade.

Nesse trabalho não se deseja construir um algoritmo de roteirização de veículos, mas sim um banco de dados relacional, mantendo a responsabilidade do gerenciamento da capacidade e da demanda aos atendentes.

Uma definição simples para banco de dados é trazida por Date (2004): “um banco de dados consiste de algumas coleções de dados persistentes que são utilizadas por aplicações de determinadas empresas e administradas por um sistema de gestão de banco de dados”.

Cada servidor de banco de dados, de acordo com Van Der Lans (2007), possui uma linguagem própria, ou mais de uma linguagem. O banco de dados trazido neste trabalho usa a *Structured Query Language* (SQL), que é uma linguagem de banco de dados *open source* usada para formular declarações e consultas em um servidor de banco de dados chamado de MySQL (VAN DER LANS, 2007). Portanto, de acordo com o autor, SQL é a linguagem nativa de um servidor MySQL, podendo ser possível determinar diferentes tipos de consultas e estratificações nos dados armazenados. Segundo o autor, os dados de um banco de dados só se tornam úteis quando uma aplicação é feita neles. Essa aplicação pode ser feita pelo sistema de gerenciamento de banco de dados (em inglês, database management system, DBMS) o qual é programado para extrair, consultar, inserir, atualizar, deletar e proteger as informações referentes. O servidor, além de realizar consultas no banco, ainda mantém a integridade dos dados, o que garante que eles estejam de acordo com a situação real proposta. Portanto, o MySQL é usado de forma interativa. O usuário entra com a declaração que deseja, o servidor processa imediatamente e o resultado é também imediatamente visto. São essenciais, portanto, ao desenvolvimento de um sistema gerencial de banco de dados (e ao SQL), o usuário, a aplicação e o servidor do banco de dados (Figura 1).

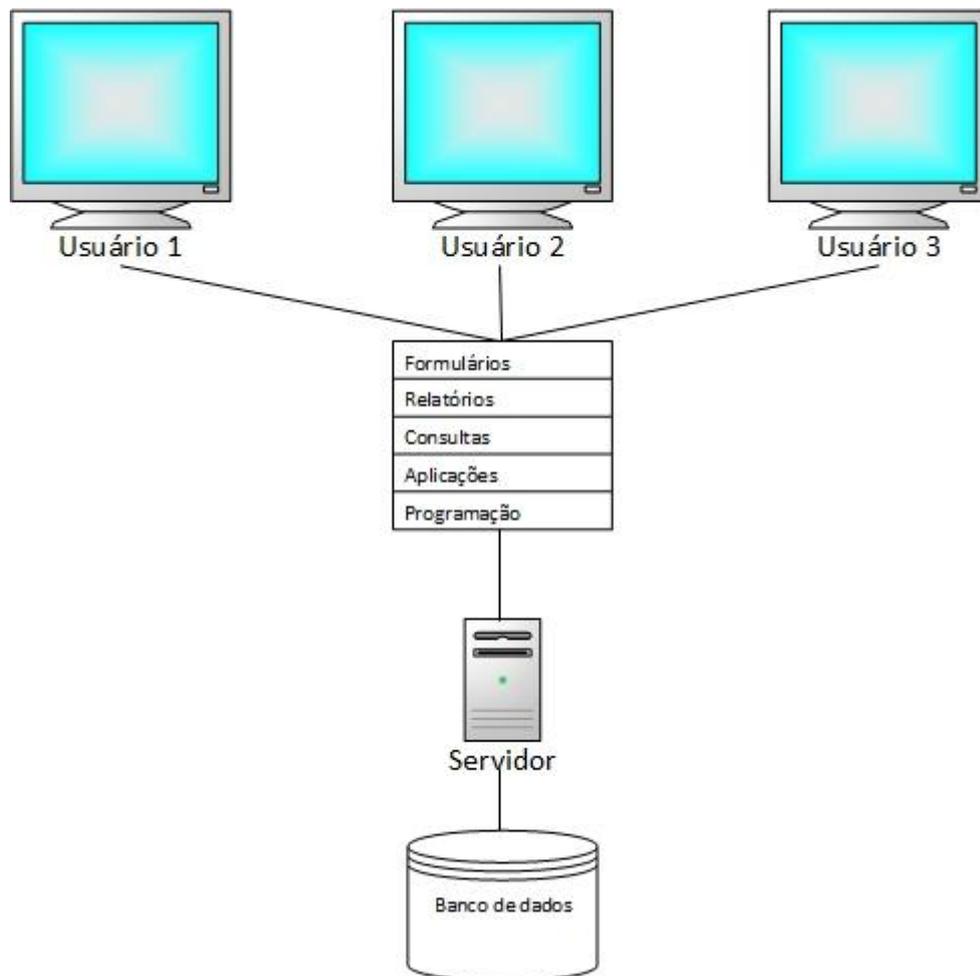


Figura 1 – Interação de um DBMS  
 Fonte: adaptado de Van Der Lans (2007)

Antes da programação em si, uma grande vantagem de um banco de dados é seu poder de armazenar as informações de modo persistente e por tempo indeterminado, além da integridade (consistência dos dados) que mantém o banco operando de acordo com as regras do mundo real, para qual o sistema foi modelado (VAN DER LANS, 2007).

Existem diversos conceitos e teorias (formais e matemáticas) quando se trata de desenvolver um banco de dados relacional (VAN DER LANS, 2007). Esses conceitos buscam organizar, relacionar e manipular os dados com os quais o usuário pretende trabalhar, e são a base ferramental de qualquer modelo de linguagem de qualquer banco de dados. Serão introduzidos tais conceitos com base em Date (2004). A iniciar, um banco de dados, como descrito anteriormente, é um conjunto de tabelas. Cada tabela é chamada de entidade no processo de modelagem do banco. Os dados são armazenados fundamentalmente em uma tabela a qual ilustra informações referentes a esse conjunto próprio de dados. Nessa tabela, cada coluna (ou campo) define uma linha. Ou seja, cada linha da tabela possui um mesmo

número de atributos, e cada linha da tabela armazena informações sobre aquela entrada. O banco de dados relaciona e conecta linhas das diferentes tabelas que o compõem através de diferentes conceitos de relação existentes. As relações efetivas e representam as regras do negócio em estudo. Para isso, são usadas chaves primárias para identificar de forma única cada entrada de uma tabela. Uma chave primária, de acordo com Date (2004), é um valor único que simboliza aquela linha, e seu valor nunca se repetirá novamente naquela tabela. Porém, na medida em que o banco de dados se desenvolve, chaves primárias podem se tornar uma chave estrangeira em outra tabela onde há relação. Estamos então, com o conceito de relação, ligando um campo de uma tabela A com um campo de uma tabela B, por exemplo.

Bagui e Earp (2011) definem relação em banco de dados como a conexão entre os registros (linhas) nas diferentes entidades. As relações são geralmente convocadas como verbos e possuem cardinalidade nas duas direções descritas pelas entidades, segundo os autores. Serão estudadas nesse trabalho relações binárias do tipo 1:1, 1:N, N:M, com base em Bagui e Earp (2011).

- Relação 1:1 (um para um): esse tipo de relação é unívoco, ou seja, uma entidade está relacionada com outra entidade, e vice-versa. Isso significa que um registro de uma tabela é ligado a outro registro de outra tabela uma única vez. A Figura 2 ilustra esse tipo de relação.

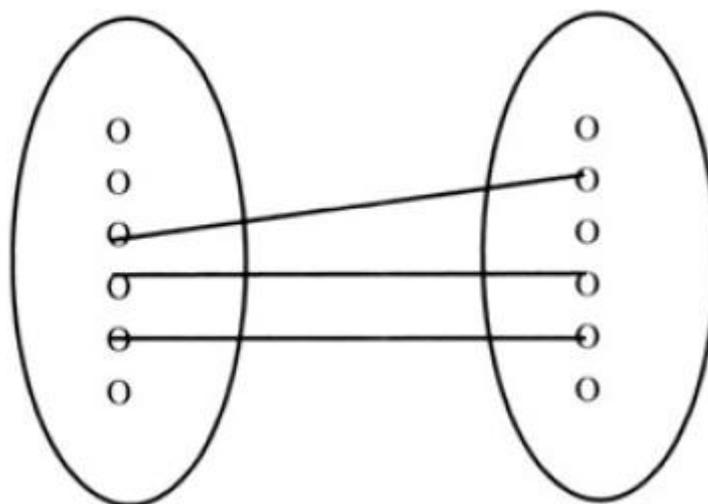


Figura 2 – Relação 1:1  
Fonte: adaptado de Bagui e Earp (2011)

- Relação 1:N (um para muitos): significa que um registro de uma tabela está relacionado com muitos outros registros da outra tabela. Os autores aqui recomendam estar claro qual das entidades recebe a cardinalidade 1 e qual receberá a cardinalidade

N. A relação N:1 (muitos para um) recebe a mesma definição. A Figura 3 traz um exemplo dessa relação.

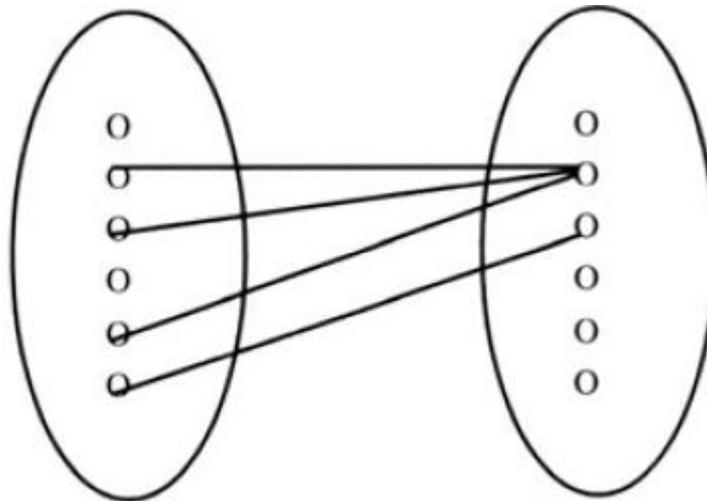


Figura 3 – Relação 1:N  
Fonte: adaptado de Bagui e Earp (2011)

- Relação N:M (muitos para muitos): aqui, muitas ocorrências na tabela A estarão ligadas com diversas ocorrências de uma tabela B. É necessário criar uma tabela auxiliar quando se encontra essa situação, a qual receberá um composto de chaves primárias das duas tabelas. Assim, essa relação se transforma em 1:N, onde o lado 1 fica com as tabelas em estudo e o lado N com a tabela auxiliar. As Figuras 4 e 5 trazem o conceito N:M, sendo que a Figura 5 traz a notação de Chen (1976), sendo a notação mais clássica para modelagem de dados.

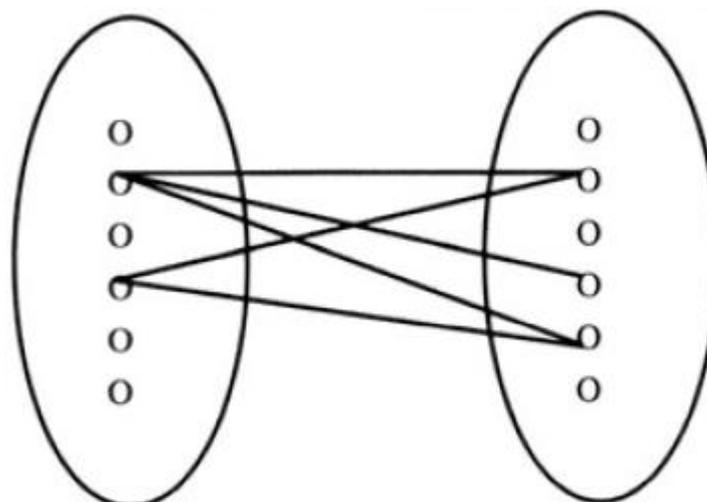


Figura 4 – Relação N:M  
Fonte: adaptado de Bagui e Earp (2011)

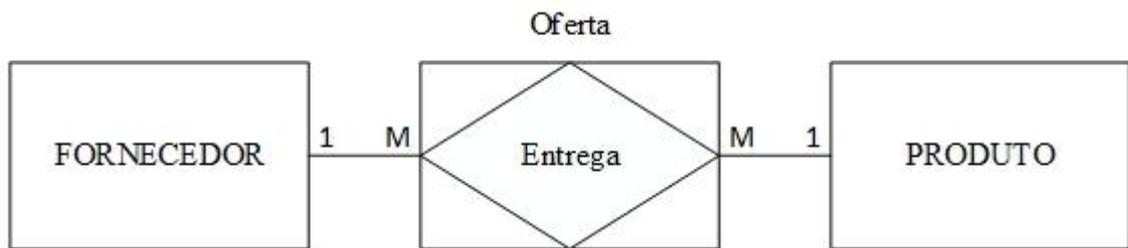


Figura 5 – Tabela auxiliar (associativa)  
 Fonte: adaptado de Kroenke, Bunker e Wilson (2010)

As relações ainda podem contar com a participação total ou parcial pelas entidades. As relações totais ou mandatórias exigem que um registro de uma tabela esteja associado a um registro de outra tabela. Já uma relação parcial (ou opcional) ocorre quando uma instancia da entidade pode ou não estar associada a um ou mais registros de outra entidade (BAGUI E EARP, 2011).

Definidos os conceitos básicos do banco de dados relacional, busca-se uma forma para iniciar a modelagem do mesmo e o seu funcionamento. Uma forma prática de ilustrar e modelar situações reais em banco de dados é o diagrama entidade-relacionamento (de agora em diante, ERD, em inglês).

## 2.2. Arranjo e adequação dos dados para representação no sistema

Os dados utilizados em um sistema gerencial também devem ser arranjados e organizados de uma forma a representar e caracterizar as necessidades do usuário. De acordo com Bagui e Earp (2011) o usuário de um sistema de banco de dados muitas vezes não possui grandes conhecimentos em informática. A chamada Engenharia de Software traz metodologias diferentes para o modelo conceitual de um banco de dados relacional. Os autores dizem que a narrativa que o usuário traz para um analista pode ser ilustrada através de um ERD, diagrama que auxilia de forma expressiva a montagem do banco de dados. O analista deve, portanto, saber ouvir o usuário e de sua narrativa modelar um sistema condizente com a realidade.

A modelagem conceitual de dados possui diversas notações diferentes.

A modelagem conceitual é sempre feita com base em algum formalismo conceitual (ex.: Entidade-Relacionamento, Orientação a Objetos). O resultado do processo de modelagem, denominado esquema conceitual, é apresentado através de uma linguagem formal de descrição que pode estar expressa através de uma sintaxe e/ou uma notação gráfica. Para cada formalismo conceitual podem existir diversas

linguagens de descrição de esquema que são compatíveis com o formalismo (EUROPEAN COMMITTEE FOR STANDARIZATION, 1996 apud JUGURTA, 2000, p. 30).

A notação utilizada no presente trabalho é a *Unified Modeling Language* (UML de agora em diante). Ela não caracteriza uma metodologia de desenvolvimento de um sistema, mas sim um padrão adotado pelo analista no momento de representar objetos e suas conexões (JUGURTA, 2000). Por ser uma notação gráfica, cabem algumas regras no desenvolvimento do diagrama de classes que servirá de base para o banco de dados relacional. Uma entidade, na notação UML é chama de classe, e possui suas características (atributos) próprias. As classes se relacionam entre si por diversas multiplicidades em relações, que lidas nos dois sentidos, associam objetos de uma classe a objetos de outra classe. Para expressar uma multiplicidade (ou cardinalidade) entre duas classes, Jugurta (2000) comenta que a notação utilizada indica quantos objetos podem estar relacionados através dessa associação. Alguns exemplos de multiplicidade: zero para um (0..1), um para vários (1..\*) e muitos para muitos (m..n). A Figura 6 ilustra um exemplo da notação UML de classes, atributos, relacionamentos e multiplicidade de forma resumida. Nos quadrados se encontram as entidades e abaixo deles, os atributos relativos a si. A flecha escura indica o sentido de leitura da relação.

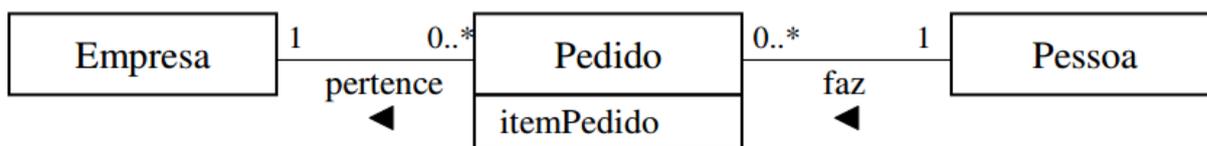


Figura 6 – Notação UML resumida  
Fonte: adaptado de Jugurta (2000)

Para a concepção do banco de dados em si, os Analistas ou Engenheiros de Software utilizam de diversos processos de desenvolvimento do banco de dados. A Figura 7 traz a metodologia utilizada para este trabalho, onde o problema surge na prática. A figura pode ser lida seguindo as setas, ou analogamente a um quadrante, sua leitura se faz dos quadrantes II a I, I a IV, IV a III. No quadrante II surge o problema conceitual, com uma narrativa geralmente associada a dados práticos que o usuário relata ao Analista ou Engenheiro de Software. Nesse primeiro passo, o estudo de caso é apresentado e inicialmente analisado. Sugere-se uma entrevista com o interessado (usuário). No quadrante I surge uma modelagem inicial do modelo conceitual do banco de dados, seguindo uma das diferentes notações existentes. No presente trabalho, a

notação utilizada é a UML, como dito anteriormente. Nessa etapa surgem os primeiros modelos de entidades (tabelas e seus atributos) e as relações entre elas. O analista aqui deve estar ciente da modelagem, de forma que ela contemple todos os requisitos do usuário. É interessante dizer que projetar o banco de dados significa também estudar o problema. O desenho do banco de dados relacional em alguma plataforma específica para modelagem surge no quadrante IV. No quadrante III a solução é apresentada pelo analista. O banco de dados completo, com dados inseridos e rotinas de consultas e formulários exigidos pelo usuário, encontra-se finalizado e pronto para uso e implementação.

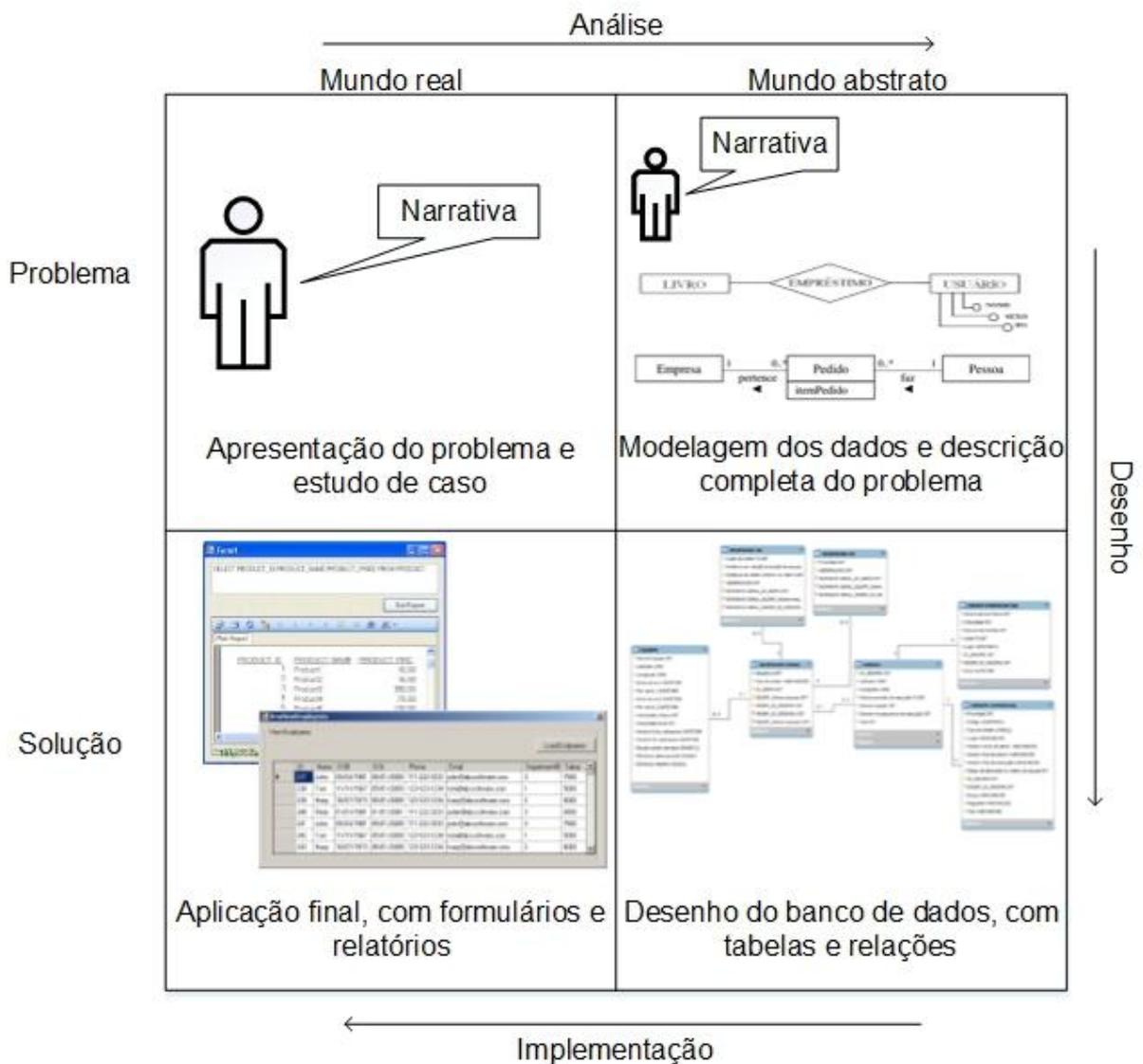


Figura 7 – Metodologia empregada  
 Fonte: adaptado de Churcher (2007)

Um banco de dados bem estruturado pode ainda servir como um sistema de suporte à decisão. Shen et al (2012) discute sobre a interface gráfica (*display*) e o arranjo de dados para usuários de centros de atendimento nos quais se trabalha com atendimento a serviços e decisões emergenciais ou críticas. Os autores relatam que na literatura há discussões sobre o aprimoramento da gestão de sistemas de informação para emergências (EMIS, em inglês).

Os EMIS ultimamente vêm ganhando espaço no que diz respeito a aplicações que requerem integração de dados, comunicação e ferramentas de interface e visualização.

A habilidade de tomar decisões de forma rápida e efetiva em situações de alteração na demanda programada, por exemplo, é extremamente importante. Os sistemas de informação estão sendo, portanto, desenvolvidos para auxiliar os serviços de atendimento a coletar rapidamente, organizar e tomar decisões efetivas sob pressão ou ainda manter os dados armazenados (SHEN et al., 2012).

Perrier et al (2013b) define um problema de gerenciamento de serviço, como o do presente trabalho, como uma decisão emergencial relacionada ao contingente e pertencente a um nível de planejamento em tempo real. Os autores ainda explicam que esse tipo de problema visa minimizar o intervalo de tempo na qual o cliente permanece sem ou no aguardo do serviço, levando em consideração os custos, distâncias, horas trabalhadas das equipes e suas cargas de trabalho (que deve ser o mais balanceado possível).

### **2.3. Contexto de gerenciamento de serviços em concessionárias de distribuição de energia elétrica**

Todo sistema de distribuição de energia elétrica no mundo nasce de um monopólio natural (BERGEY, RAGSDALE E HOSKOTE, 2003). A eletricidade é uma *commodity*, que como tal pode ser comprada e vendida em um mercado livre. De acordo com os autores, a palavra distribuição nesse contexto se interpreta como o processo de redução de voltagem adquirida na geração e a entrega da energia elétrica na localidade desejada. Já Perrier et al (2013a) esclarecem que um sistema de distribuição elétrica entrega aos clientes energia através da geração, transmissão e distribuição desde a planta geradora até subestações, que dividem a eletricidade em transformadores, reduzindo a voltagem até a casa dos clientes do sistema, conforme ilustra a Figura 8.

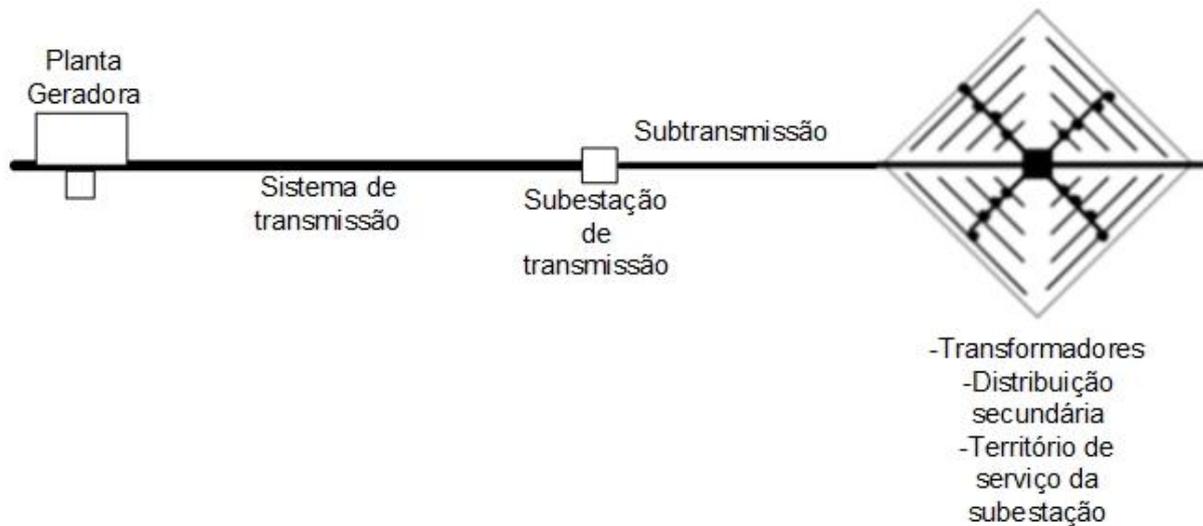


Figura 8 – Sistema de distribuição de energia elétrica  
 Fonte: Perrier (et al, 2013a)

O problema de governos de países com dificuldade em criar mercados competitivos de distribuição de energia elétrica é bastante comum ao redor do globo (BERGEY, RAGSDALE E HOSKOTE, 2003). Dentro desse problema, ainda surge a questão do distritamento da distribuição: definir, com base em certos critérios, a rede de distribuição, seu tamanho e localização de nódulos de clientes. Bergey, Ragsdale e Hoskote (2003) citam a importância dessa definição, que se dá no fato de que, se dimensionados de forma errada, o nível de serviço provido pela concessionária será afetado. Serviços comerciais e de manutenção desses nódulos poderão, então, acarretar problemas de baixo desempenho de serviço e dificuldades na manutenção e na concessão de energia (PERRIER et al, 2013a).

O gerenciamento de serviços possui diversas denominações na literatura. Um estudo de caso trazido por Mendonza, Medaglia e Velasco (2009), traz um sistema de suporte à decisão para o circuito de água e esgoto da cidade. Esse sistema pode ser estudado como análogo a um sistema de distribuição de energia elétrica. A empresa distribuidora divide a capital em cinco zonas de serviço independentes. Cada zona é dividida em porções contendo um agrupamento de clientes que usam do mesmo circuito hidráulico. As rotas de serviço para atendimento dos clientes são manualmente distribuídas para cada atendente, buscando um balanço de carga de trabalho equilibrado para cada um. Essa roteirização pode ser mais bem executada por sistemas de informação que podem realizar o despacho dos auditores para as zonas: as ferramentas para roteirização dos veículos, que é um sistema de suporte à decisão que pode maximizar a roteirização dos processos de atendimento das ordens de serviço. Esse tipo de configuração é conhecido na literatura como problema de roteirização de veículos com

distância restrita. Os autores ainda definem outro tipo de configuração, no qual o cliente aguarda a visita do auditor (tal visita pré-agendada) como um problema de roteirização de veículos com janela de tempo. Nesse estudo de caso, os autores buscam configurações de algoritmos que tragam soluções que satisfaçam a restrição de distância.

Dependendo das necessidades, um banco de dados relacional, portanto, pode ser substituído por outros sistemas de informação com algoritmos relacionados. Esse tipo de Sistema de Informação geralmente custa mais caro para as empresas, embora possa possuir mais funcionalidades do que consultas a banco de dados, como o SQL permite.

### **3. Procedimentos metodológicos**

#### **3.1 Definição do contexto de aplicação: gerenciamento de serviços em concessionárias de distribuição de energia elétrica**

Este trabalho, aplicado em uma concessionária de distribuição de energia elétrica no Rio Grande do Sul (responsável por atender uma área de mais de 99 mil quilômetros quadrados no estado), estuda o processo de despacho dos serviços dessa área, que é dividida em cinco postos técnicos que gerenciam as demandas em áreas pré-estabelecidas.

Como dito anteriormente, o processo de atendimento e programação das ordens de serviço ocorre de duas formas: comercial/técnica e emergencial. Ordens emergenciais, aquelas com mais urgência no reparo, se sobrepõem às ordens de serviços comerciais para a equipe designada. Desse modo, acontecem variações na execução de ordens pré-selecionadas para o dia, podendo ocorrer postergamento de ordens comerciais.

O serviço prestado possui um prazo de atendimento de acordo com determinação do órgão regulador das concessionárias Agência Nacional de Energia Elétrica (ANEEL). A alocação de uma ordem de serviço a uma equipe não garante a sua execução, justamente por causa do fator citado acima. Ainda, a regulação da ANEEL, que atinge alguns tipos de ordem, exige da concessionária a priorização no atendimento desses tipos de ordens, de modo a evitar multas e custos indesejados.

### **3.2 Banco de dados para o gerenciamento de serviços**

O banco de dados relacional se torna uma ferramenta bastante interessante no momento de sistematizar e estruturar o atendimento dos serviços pela concessionária. Através da narrativa do problema, ele se adequa e se ajusta às diferentes demandas práticas, além de ser um conceito *open source*. Isso garante seu uso bastante difundido em gerenciamento de serviços e sua vasta aplicação.

Utilizando a metodologia apresentada em 2.2 (Figura 7), os dados práticos obtidos puderam ser adequados a um modelo de banco de dados, estruturando seu despacho e garantindo a integridade dos dados, além de garantir o armazenamento dos mesmos.

No que se refere à classificação da pesquisa, esse estudo de caso se especifica como de natureza aplicada, tendo em vista que o resultado será um sistema em linguagem SQL que formará um sistema de suporte à decisão, gerando informações a partir de dados. A abordagem é uma combinação de tópicos quantitativos e qualitativos, de campo exploratório.

Foram programadas seis etapas para o trabalho: (i) estudo da literatura: leitura e análise das referências bibliográficas, (ii) estudo da natureza do problema de gerenciamento de atendimento: como transformar os dados existentes em informação para suporte à decisão, (iii) definição do modelo de dados a ser adotado: modelagem do banco de dados e testes com os dados reais obtidos, (iv) construção de indicadores para mostrar o desempenho do gerenciamento: simulações com dados existentes, (v) estudo de caso modelando esses casos referidos: consolidação do banco de dados e estruturação das relações dos dados e (vi) avaliação e conclusões: avaliação do sistema e sua adequação à realidade do problema.

Para a concepção do banco de dados em si, e a sua sincronização com um servidor privado, foram utilizados softwares livres como o *MySQL Workbench 6.3 CE* e o *phpMyAdmin*. Essas plataformas são de fácil entendimento e organização e possuem vasta referência de instruções.

## **4. Resultados e discussão**

O principal resultado a ser atingido com a estruturação do banco de dados se resume em relacionar capacidade com demanda. Capacidade, nesse estudo, caracteriza as equipes de atendimento dos serviços (recursos internos). Já a demanda, como dito anteriormente, caracteriza as ordens comerciais e emergenciais, vindas dos clientes pertencentes a uma determinada região atendida pela concessionária. Portanto, a associação e vinculação de

equipes às ordens (despacho) se torna importante agora, na modelagem conceitual do banco de dados relacional. É de extrema importância a diferenciação entre ordens emergenciais e comerciais. Isso porque cada uma possui características (e atributos) diferentes, mesmo que alguns campos são comuns a todas as ordens. A solução de despacho é também chamada de calendário, o qual registra conjuntos de equipes a conjuntos de ordens. Uma última característica do problema em estudo é importante antes do início da modelagem: uma ordem pode aparecer em uma solução de despacho mais de uma vez. Sua entrada em um despacho não significa que ela será atendida com toda certeza, visto que a sequência das equipes no dia varia de acordo com a entrada de novas ordens emergenciais. Por isso, uma ordem pode aparecer em diversos despachos.

A forma desestruturada que os dados práticos obtidos diretamente da concessionária se apresentaram caracteriza o primeiro desafio no desenvolvimento da metodologia e concepção do banco de dados relacional. Os dados, recebidos em planilhas, tiveram que ser organizados de forma a estabelecer as entidades (classes, pela notação UML), buscando modelar as relações entre si. Essa etapa corresponde aos quadrantes II e I da metodologia apresentada na Figura 7 (sessão 2.2), a qual significa, a partir de uma narrativa, a concepção inicial e o modelo conceitual do diagrama entidade-relacionamento (ERD).

Foram identificadas, inicialmente, três entidades: Equipe, Ordem e a associação entre os dois (Despacho). Na medida em que a modelagem do ERD avança, surge a necessidade de criar outras entidades auxiliares, de forma a tornar o modelo conceitual o mais próximo do problema real relatado. As ordens, por exemplo. Uma ordem pode ser uma ordem comercial ou pode ser uma ordem emergencial. Destaca-se a palavra “ou” na frase anterior. “ou”, nesse contexto, significa que uma ordem só pode ser comercial ou emergencial, de forma unívoca. Portanto, na modelagem em notação UML, criaram-se outras duas tabelas para a entidade Ordem. A Figura 9 ilustra o caso dessa entidade, que se desdobra em outras duas tabelas: Ordem Comercial e Ordem Emergencial. Os atributos, vindos dos dados obtidos, foram também divididos entre as três tabelas, evitando redundâncias.

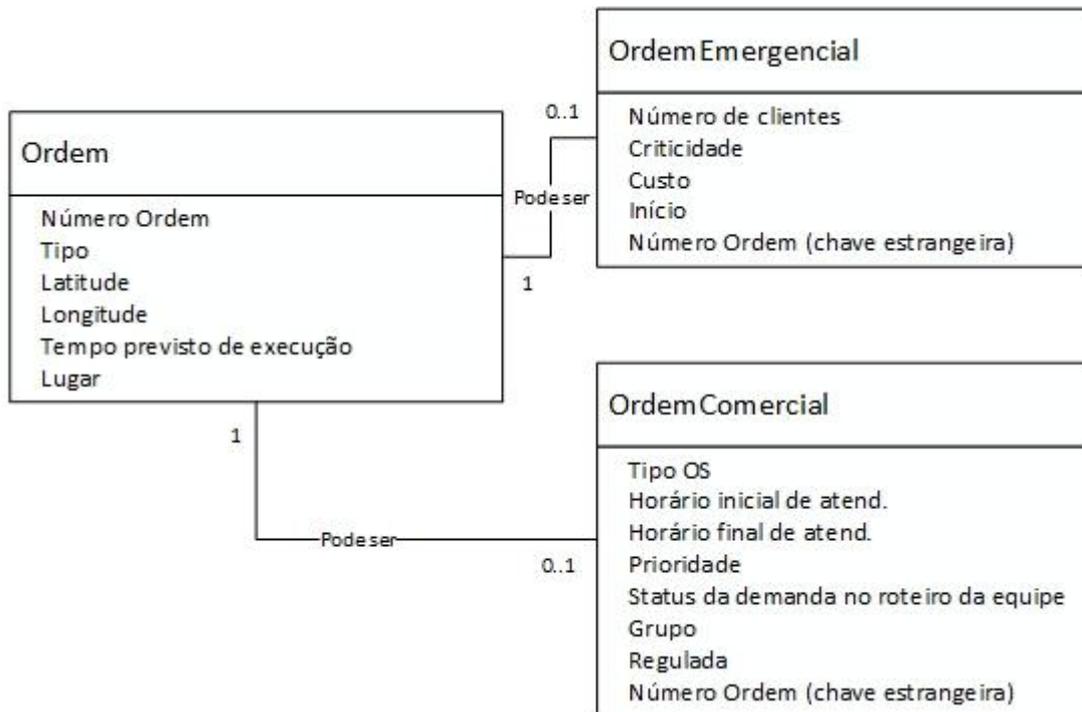


Figura 9 – Modelagem do caso das ordens  
 Fonte: desenvolvido pelo autor

Foi configurada uma relação 1:1 entre a tabela Ordem e as tabelas Ordem Comercial e Ordem Emergencial, sendo que a cardinalidade das duas novas tabelas não é mandatória (devido à existência da palavra “ou”). Isso torna a notação 1:1 para 1:0..1, sendo o lado 0..1 não mandatório, se adequando à realidade (uma ordem pode ser uma ordem comercial ou pode ser uma ordem emergencial). Ainda, as classes OrdemComercial e OrdemEmergencial recebem, portanto, Número Ordem como uma chave estrangeira, garantindo a relação entre as tabelas.

O mesmo caso das ordens ocorre para a entidade associativa Despacho. Um despacho pode ser um despacho comercial ou pode ser um despacho emergencial. Ou ainda, um despacho geral é caracterizado por um despacho comercial ou um despacho emergencial. Essa frase é absoluta, e advém da narrativa inicial do problema em estudo. A Figura 10 elucida a relação entre os despachos.

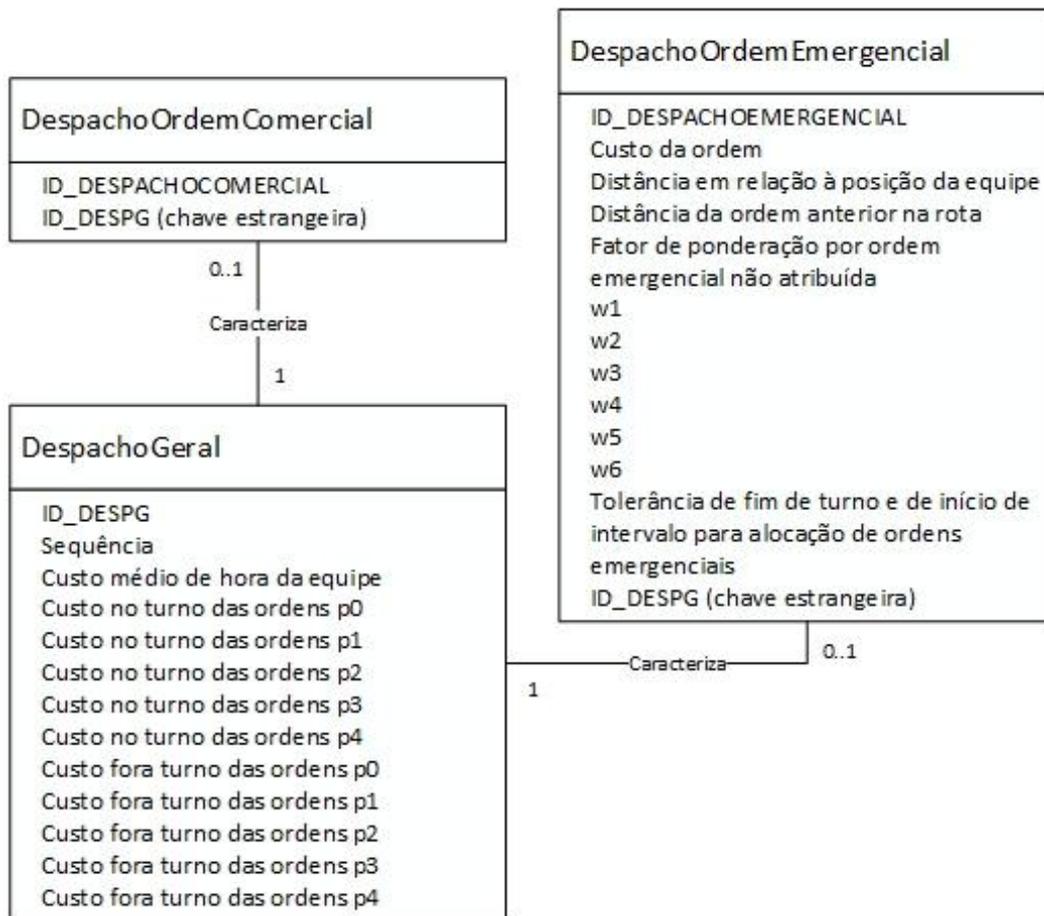


Figura 10 – Modelagem do caso do despacho  
 Fonte: desenvolvido pelo autor

As equipes (entidade Equipe) ingressam no ERD se relacionando com o Despacho Geral. Da narrativa, modela-se que as equipes estão em nenhum ou em muitos despachos. Um despacho sempre terá uma equipe associada (Figura 11). Por configurar uma relação N:M, a tabela associativa Equipe\_DespachoGeral foi criada.

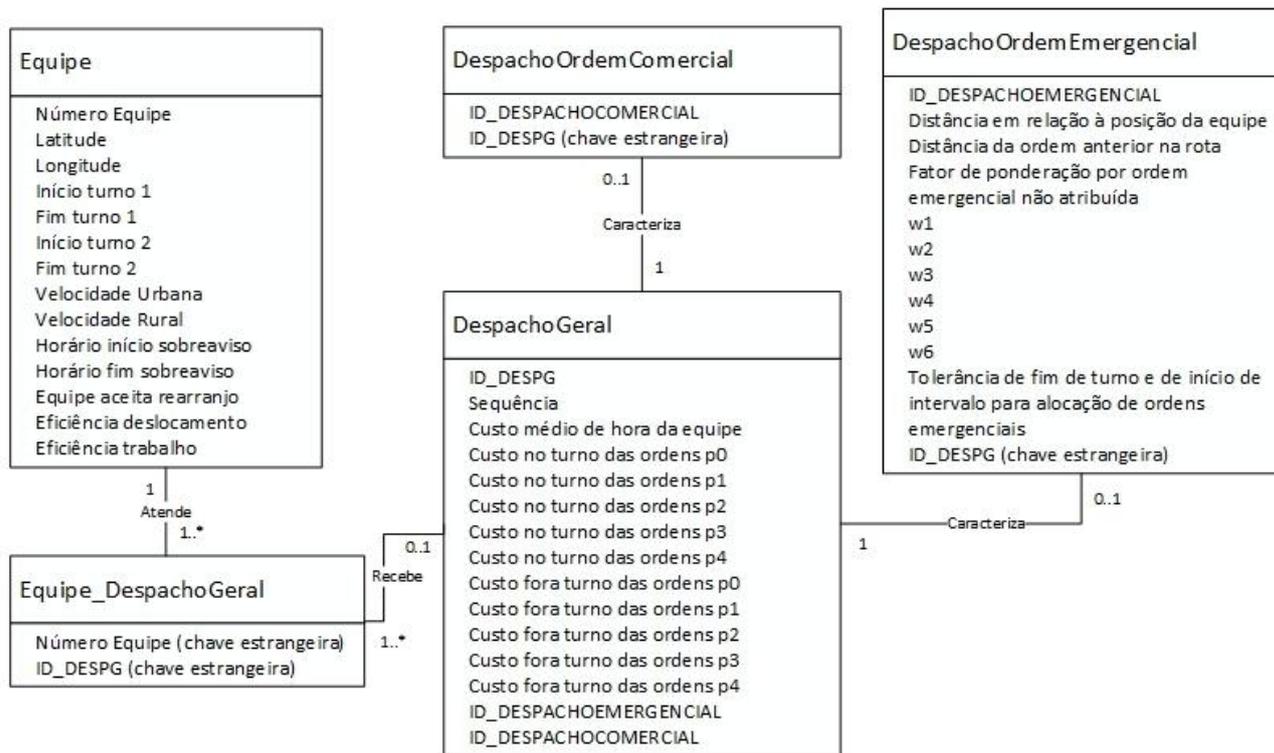


Figura 11 – Modelagem das equipes com o despacho  
 Fonte: desenvolvido pelo autor

Dando sequência na modelagem do ERD, representando o quadrante I da Figura 7, resta agora relacionar as ordens à tabela do despacho geral. Da natureza do problema, se relata que um despacho possui uma ou muitas ordens (comerciais e emergenciais) associadas. Ainda, cada ordem está em 0 ou muitos despachos, caracterizando uma relação N:M. Surge, então, a necessidade de uma tabela associativa (como exemplo à Figura 5) de modo a relacionar os despachos corretamente com as ordens. A tabela *Despacho\_Ordem* surge da relação N:M. O Apêndice 1 conclui a construção do ERD, totalizando nove diferentes tabelas que surgiram da ideia inicial do problema (relação entre capacidade e demanda, ou relação entre equipes, ordens e seus despachos). Encerra-se nesse momento também a concepção do quadrante I, formando o desenho inicial do banco de dados relacional.

Vale a pena comentar sobre alguns atributos existentes. Eles não foram indicados pelo autor, mas sim vindos dos dados práticos obtidos. Na tabela *Equipe*, por exemplo, os campos eficiência deslocamento e eficiência trabalho são atributos da equipe advindos de parâmetros do sistema, e variam de 0 a 1000. Na tabela *DespachoGeral*, o campo sequência indica às equipes o seguimento das rotas das ordens. Em *DespachoEmergencial*, os campos w1 a w6 indicam percentuais (de 0 a 1) de forma a formar uma ponderação em diferentes critérios. Por

exemplo:  $w_3$  representa a ponderação sobre o tempo de atraso, depois do final de turno, causado pelo adiamento da execução das ordens comerciais de todas as prioridades. Já  $w_5$  representa a ponderação sobre o tempo total de deslocamento considerando todas as ordens, sejam comerciais ou emergenciais e  $w_6$  representa a ponderação sobre o cancelamento daquela ordem emergencial. As tabelas de ordem, ordem emergencial e ordem comercial possuem atributos únicos de estruturação e por isso tiveram que ser separadas. As ordens emergenciais possuem um campo de criticidade, que pode ser 0 (não há perigo), 1 (há perigo) e 2 (cliente importante). A tabela Ordem, que define todas as ordens, possui o campo Tempo previsto para execução e lugar (U de urbana ou R de rural). O campo Tipo (C ou E, comercial ou emergencial) define se a ordem registra na tabela OrdemComercial ou na tabela OrdemEmergencial. A tabela de ordens comerciais contém o campo Grupo (AES ou CLIENTE). AES indica que a ordem comercial é técnica, e CLIENTE indica que a ordem parte de algum cliente. O campo Regulada indica se a ANEEL fiscaliza ou não o atendimento da ordem vinda dos clientes. De fato, as ordens reguladas pela ANEEL possuem maior peso na hora do despacho do que as não reguladas legislativamente. O campo Tipo OS define o campo Tempo previsto de execução, da tabela Ordem. Esse tempo é importante no momento de estimar quantas ordens poderão ser atendidas no despacho (sem a possibilidade de inserção de ordens emergenciais).

Com o ERD completo, e seu modelo lógico condizente com a realidade do problema, é possível iniciar o desenho do mesmo em um software específico para construção e concepção de banco de dados, que use linguagem SQL para formular as declarações e MySQL como servidor do banco de dados. O software utilizado para a concepção do banco foi o *MySQL Workbench*. A concepção do mesmo pelo software segue exatamente o ERD, indicando o tipo de dado que será inserido em cada campo. Dentre as funcionalidades do software está a possibilidade de criar o banco de dados de um modo bastante *user-friendly*, o que facilita sua concepção. Além disso, o software gera o *script* (as linha de comando utilizadas em linguagem SQL) automaticamente por trás da interface, podendo facilmente ser exportado em formato .sql ou similar a qualquer momento. Com o script (Apêndice 3), é possível replicar o banco de dados (este vazio, sem os dados inseridos), utilizando os mesmos comandos, por ilimitadas vezes.

A metodologia segue, portanto, através do quadrante IV, a concepção do banco no MySQL Workbench. O Apêndice 2 ilustra o ERD completo, reproduzido de forma igual à modelagem realizada. Sobre o tipo de dado escolhido (item que acompanha o nome do atributo), nesse

trabalho se utilizou INT para os números, VARCHAR para campos que exigiam letras, MEDIUM TEXT para campos de latitude e longitude e DATETIME e TIME para os campos que exigem valores de datas e horários ou somente horários.

A sincronização do banco de dados com um servidor garante que o sistema possa ser acessado de qualquer computador que possua um software de navegação na internet (*browser*). O servidor também permite que o banco de dados possa ser acessado e modificado simultaneamente por diversos usuários, tolerando a propriedade de isolamento do banco de dados. Para fins de esclarecimento, o banco de dados desenvolvido nesse trabalho foi sincronizado em um servidor privado durante a sua concepção e uso.

Os dados práticos das equipes e das ordens (comerciais e emergenciais) foram ajustados, portanto, de modo a serem inseridos no banco de dados já pronto. Quando no término da sua construção, ele se apresenta vazio, pronto para armazenar tantos dados quanto forem necessários. Para inserir os dados nas tabelas, foi utilizado um assistente de importação de dados que acompanha o aplicativo *phpMyAdmin*. Também foi gerado um programa em linguagem SQL (*script*) para que os mesmos dados possam ser inseridos novamente quando o banco for replicado (Apêndice 4). Para a inserção das equipes, foram conseguidos dados referentes a três equipes. Alguns campos permaneceram nulos. Os dados já inseridos podem ser visualizados na Figura 12. A tabela EQUIPE possui agora três registros.

Número equipe	Latitude	Longitude	Início turno 1	Fim turno 1	Início turno 2	Fim turno 2	Velocidade Urbana	Velocidade Rural	Horário Início sobreaviso	Horário Fim sobreaviso	Equipe aceita rearranjo
237	29890370000	51135880000	2015-03-17 18:00:00	2015-03-17 22:00:00	2015-03-18 13:00:00	2015-03-18 17:00:00	0	0	NULL	NULL	0
283	29899710000	51140690000	2015-03-17 17:00:00	2015-03-17 21:00:00	2015-03-18 12:00:00	2015-03-18 16:00:00	0	0	NULL	NULL	0
285	29889090000	51131540000	2015-03-17 17:00:00	2015-03-17 22:00:00	2015-03-18 13:00:00	2015-03-18 16:00:00	0	0	NULL	NULL	0

Figura 12 – Dados de equipes  
Fonte: desenvolvido pelo autor

O mesmo procedimento de inserção de dados ocorreu com as ordens (Apêndice 4). Foram inseridas 180 ordens, sendo 90 ordens comerciais e 90 ordens emergenciais, retiradas de um período de pouco mais de uma semana. Portanto, a tabela Ordem possui 180 registros, enquanto que as tabelas OrdemComercial e OrdemEmergencial possuem 90 registros cada uma. Percebe-se a relação entre essas, como representado pela Figura 9, através da chave estrangeira Número Ordem.

Com os dados de equipes e ordens inseridos é possível criar os despachos gerais. Para isso, cria-se registros na tabela *DespachoGeral* (ver o *script* de inserção no Apêndice 4) e a partir deles, se relaciona, manualmente através do servidor, equipes e ordens.

Foram gerados cinco despachos fictícios, entre despachos comerciais e emergenciais, para dividi-los entre as equipes. Optou-se por não utilizar todas as 180 ordens existentes no banco de dados, para manter os despachos condizentes com a realidade do problema. Depois de criado os despachos gerais (de 1 a 5), se inserem os mesmos nas tabelas de despachos comerciais e despachos emergenciais, fazendo a divisão entre despachos. Lembra-se que uma equipe recebe, diariamente, despachos comerciais pré-programados, enquanto que os despachos emergenciais surgem no transcorrer do turno. Escolheu-se por atribuir os despachos gerais 1 e 2 aos despachos emergenciais 200 e 201 respectivamente. Os despachos gerais 3, 4 e 5 são referenciados pelos despachos comerciais 102 , 100 e 101, respectivamente. A Tabela 1 ilustra os despachos feitos, sendo que os números seguem a ordem *ID\_DESPG (ID\_DESPACHOEMERGENCIAL)* ou *ID\_DESPG (ID\_DESPACHOCOMERCIAL)*. Ou seja, o número de 1 a 5 que aparece fora dos parênteses deriva da tabela *DespachoGeral*, enquanto que o número que se encontra dentro dos parênteses decorre das tabelas *DespachoOrdemEmergencial* e *DespachoOrdemComercial*. A Figura 13 apresenta o problema de forma didática e ilustrativa.

Equipe	237	263	265
Despacho Ordens emergenciais	1 (200)	2 (201)	
	4527672	4527900	
	4527818	4527932	
Ordens atribuídas	4527835	4527986	
	4527864	4527989	
	4527898	4528006	
Despachos Ordens comerciais	4 (100)	5 (101)	3 (102)
	85024033	85024241	85024415
	85024077	85024243	85024432
	85024082	85024261	85024433
	85024087	85024264	85024443
Ordens atribuídas	85024093	85024265	85024448
	85024094	85024315	85024476
	85024095	85024327	85024477
	85024099	85024345	85024478
	85024118	85024348	85024479
	85024126	85024385	85024482

Tabela 1 – Despachos  
Fonte: desenvolvido pelo autor

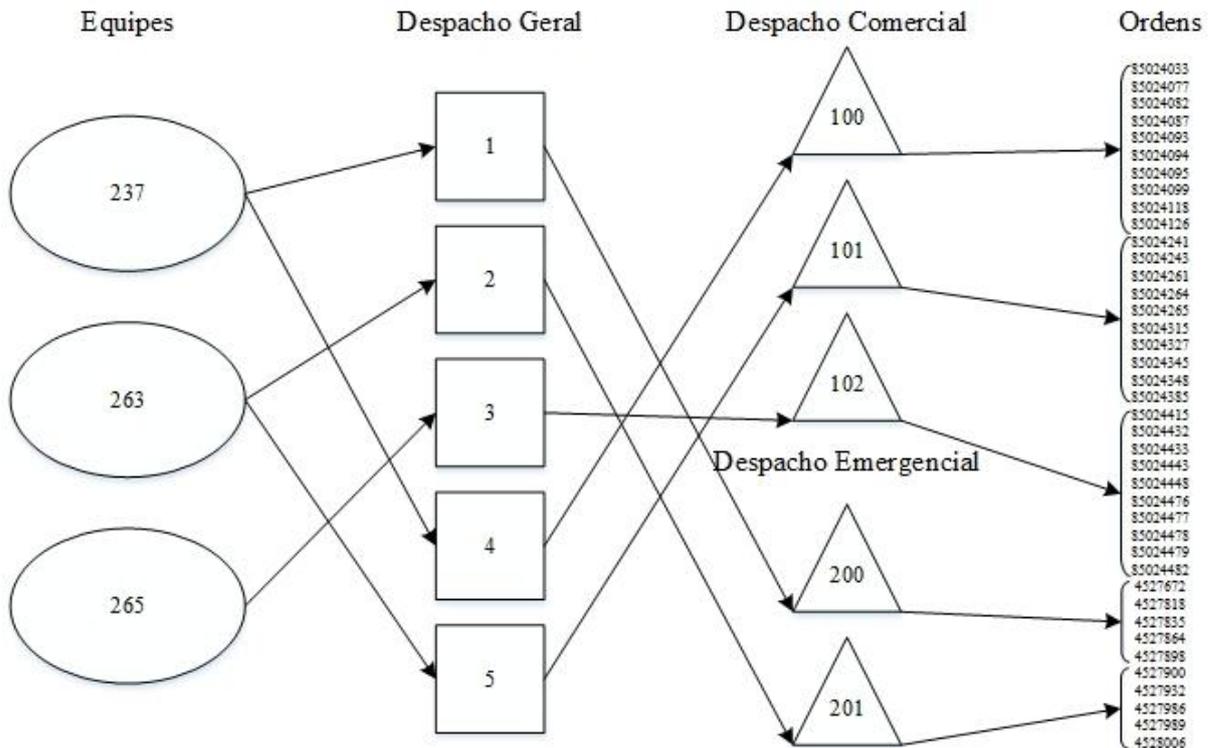


Figura 13 – Visualização dos despachos  
Fonte: desenvolvido pelo autor

Com os despachos criados, resta agora associar equipes e ordens aos mesmos no banco de dados. Para isso, existem as tabelas associativas criadas entre as tabelas de equipes e ordens e a tabela de despacho geral. Essas tabelas associativas foram nomeadas Equipe\_DespachoGeral e Despacho\_Ordem (ver Apêndice 1) e possuem como atributos as chaves estrangeiras ID\_DespachoGeral, Número Equipe e Número Ordem. Portanto, a associação das equipes às ordens ocorre através do ID\_DespachoGeral, que também é referenciado pelas tabelas de despachos comerciais e despachos emergenciais. Criando registros nessas tabelas associativas, a relação entre demanda e capacidade de fato acontece. A ilustração da Figura 13 mostra o problema de forma indireta (mas condizente com a realidade), porque o que acontece no banco de dados é a inserção direta dos despachos gerais com as ordens. Isso significa que foram inseridos 40 registros na tabela Despacho\_Ordem, associando os cinco despachos gerais com as 40 ordens selecionadas para o despacho. O *script* para a inserção desses registros encontra-se também no Apêndice 4.

A solução para o problema inicial e a implementação do sistema com as consultas já prontas (quadrante III da Figura 7) deve representar a realidade de um despacho da concessionária. É possível, como dito, realizar diversos tipos de consultas com os dados de um banco, através de linhas de comando em SQL. Pelo fato de os dados práticos não estarem completos, as consultas criadas neste trabalho são bastante simples, de forma a ilustrar o funcionamento e as possibilidades que um sistema como o apresentado possui. O que as linhas de comando fazem, na prática, com o sistema é filtrar um subconjunto de linhas, um subconjunto de colunas ou ambos, trazendo na interface o resultado desejado. É, em resumo, uma forma de perguntar uma informação à base de dados e, interativamente, obter a resposta do sistema. O software *phpMyAdmin* (2015) possui um sistema para criação de consultas de fácil uso, gerando o *script* da consulta em seguida. Foram, então, criadas consultas padrão para a verificação de alguns atendimentos.

A primeira consulta criada seleciona dos dados quais os despachos feitos em um determinado dia. Isso relembra o conceito de calendário, onde as equipes se relacionam com as ordens. Portanto, o resultado da consulta traz três colunas: Número da equipe, ID do despacho e Número das ordens atribuídas. O resultado da consulta no sistema traz os despachos programados (ficticiamente). O Apêndice 5 traz o *script* para essa consulta. O software utilizado permite exportar os resultados das consultas (Figura 14). A Figura 14 traz apenas as

primeiras linhas da consulta, por possuir 40 linhas no total. O resultado condiz com a realidade do problema e com a Tabela 1. Executando o *script*, o banco de dados retorna tais resultados.

ORDEM_Número Ordem	EQUIPE_Número equipe	ID_DESPG
4527672	237	1
4527818	237	1
4527835	237	1
4527864	237	1
4527898	237	1
4527900	263	2
4527932	263	2
4527986	263	2
4527989	263	2
4528006	263	2
85024415	265	3
85024432	265	3
85024433	265	3

Figura 14 – Resultado da consulta (mostrando a primeira página apenas)  
 Fonte: desenvolvido pelo autor

Uma segunda consulta foi elaborada, objetivando extrair do sistema a informação de qual o somatório de horas de serviço que uma equipe terá no determinado dia. Isso é possível através do campo Tempo previsto de execução, que se encontra na Tabela Ordem. Somando os registros dessa coluna para cada ordem que as equipes recebem, se tem o resultado de quantos minutos de serviço a equipe receberá com o despacho programado. Da natureza dos dados recebidos, se tem que a coluna Tempo previsto de execução possui como unidade minutos. A Figura 15 ilustra os resultados obtidos do sistema. Infelizmente, dos dados práticos obtidos, as ordens emergenciais não trouxeram valor neste campo (*null*). Portanto, a consulta trouxe somente o tempo previsto para as ordens comerciais. Vale lembrar que esse somatório não leva em consideração o tempo de deslocamento das equipes até as coordenadas geográficas das ordens.

ID Equipe	Tempo previsto de execução
237	300
263	300
265	300

Figura 15 – Resultado da consulta  
Fonte: desenvolvido pelo autor

Como foram atribuídas dez ordens emergenciais para cada equipe, sendo que cada ordem possuía um tempo previsto de execução de 30 minutos, o resultado (300 minutos) condiz com a realidade e garante que o *script* utilizado está correto. O *script* para essa consulta se encontra no Apêndice 5.

Pelas restrições que os dados práticos obtidos apresentaram, não trazendo todos os atributos necessários para completar o sistema, a criação de consultas se tornou um pouco limitada. É interessante dizer que essas consultas poderiam ser muito mais expandidas e trabalhadas, gerando uma grande quantidade de informação para o usuário e auxiliando na tomada de decisão por parte dos centros de atendimento. Ainda, o software permite criar *dashboards* mostrando essas informações de forma interativa e acessível ao usuário.

Os resultados obtidos do sistema condizem com a realidade dos centros de operação, gerando um calendário de despachos e atribuindo equipes às ordens, um dos objetivos iniciais do trabalho. A análise da relação entre capacidade e demanda foi um pouco prejudicada devido aos dados obtidos estarem incompletos. Foram obtidos dados práticos das ordens em um período de mais de um ano (porém foram inseridos no sistema dados referentes a uma semana de serviço), enquanto que somente três equipes foram inseridas na tabela Equipe. Na prática, o centro de atendimento em estudo possui entre dez e onze equipes que recebem os despachos diariamente. Percebe-se um desequilíbrio entre capacidade e demanda, sendo a demanda quase todos os dias maior que a capacidade dos centros de atendê-las.

## 5. Conclusão

Considera-se um sistema de banco de dados como o apresentado no texto uma excelente ferramenta de suporte à decisão. Os objetivos foram, de fato, atendidos com o sistema aqui apresentado. Banco de dados e suas aplicações para sistemas de informação e sistemas de

suporte à decisão possuem vasta possibilidade de modelagem de variáveis e funcionalidades que não entraram no escopo do trabalho.

Gerenciamento de serviços como os de atendimento feito por centros de operação caracterizam diversas vezes a formação do despacho, como o definido neste trabalho. A relação entre demanda e capacidade, como mostrado aqui, pode ser muito bem representada por um banco de dados relacional. Ainda, o sistema pode ser mais bem desenvolvido em questões de consulta e indicadores.

O problema de relacionar demanda com capacidade foi, na medida do possível, atendido com o banco de dados, que pode também suportar base de dados semelhantes, em problemas de gerenciamento de serviços como o apresentado. Para isso, algumas pequenas alterações poderão ser feitas, mas a base relacional pode ser mantida.

No escopo das diversas áreas da Engenharia de Produção, a modelagem e solução de problemas reais é realizada por meio softwares de planilhas eletrônicas. Na prática, em função da dimensão e do tratamento mais apurado que pode ser necessário, o uso de um banco de dados relacional para representação e armazenagem dos dados pode ser útil e eficiente.

Sugere-se, para estudos futuros, a integração do banco de dados desenvolvido com interfaces para consulta e manipulação de dados geograficamente representados.

## Referências bibliográficas

BERGEY, P. K.; RAGSDALE, C. T.; HOSKOTE, M. A decision support system for the electrical power districting problem. **Decision Support Systems**, v. 36, n. 1, p. 1-17, 2003.

BAGUI, S.; EARP, R. **Database design using entity-relationship diagrams**. CRC LLC, 2000 N.W. Corporate Blvd. Boca Raton, Florida 33431. 2011.

CHEN, P. P. S. **The entity-relationship model—toward a unified view of data**. ACM Transactions on Database Systems (TODS), v. 1, n. 1, p. 9-36, 1976.

CHURCHER, C. **Beginning Database Design. From Novice to Professional**. 2. ed. Apress. New York. 2007

DATE, C.J. **Introdução a sistemas de banco de dados**. C. J. Date ; tradução de Daniel Vieira; revisão técnica Sergio Lifschitz. 8. ed. 22 tiragem. Rio de Janeiro, RJ. Elsevier. 2004.

EUROPEAN COMMITTEE FOR STANDARDIZATION. **Geographic Information – Data Description – conceptual Schema Language**. Brussels. 1996.

KROENKE, D.; BUNKER, D.; WILSON, D. **Experiencing Mis**. Pearson Higher Education . Australia. 2013.

MENDOZA, J. E.; MEDAGLIA, A. L.; VELASCO, N. An evolutionary-based decision support system for vehicle routing: The case of a public utility. **Decision Support Systems**, v. 46, n. 3, p. 730-742, 2009.

MySQL: MySQL Workbench 6.3 Enhanced Data Migration. Disponível em: <https://www.mysql.com/products/workbench/>. Acesso em: 10/11/2015.

PERRIER, N., AGARD, B., BAPTISTE, P., FRAYRET, J., PELLERIN, R., RIOPEL, D., TRE, M. A survey of models and algorithms for emergency response logistics in electric distribution systems. Part I: Reliability planning with fault considerations. **Computers & Operations Research**, v. 40, n. 7, p. 1907-1922. 2013a

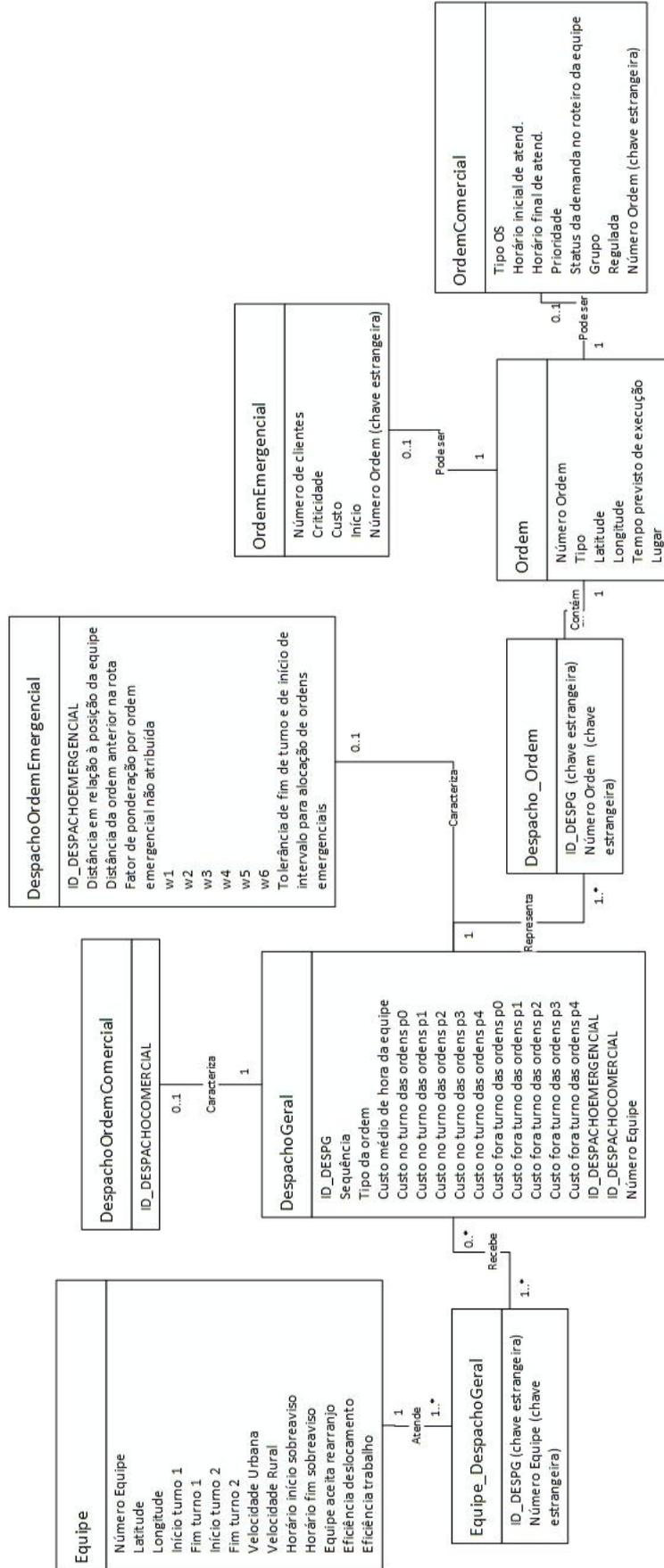
PERRIER, N., AGARD, B., BAPTISTE, P., FRAYRET, J., PELLERIN, R., RIOPEL, D., TRE, M. A survey of models and algorithms for emergency response logistics in electric distribution systems. Part II: Contingency planning level. **Computers & Operations Research**, v. 40, n. 7, p. 1907-1922. 2013b

PHPMYADMIN. Bringing MySQL to the web. Disponível em: <https://www.phpmyadmin.net/>. Acesso em: 10/11/2015.

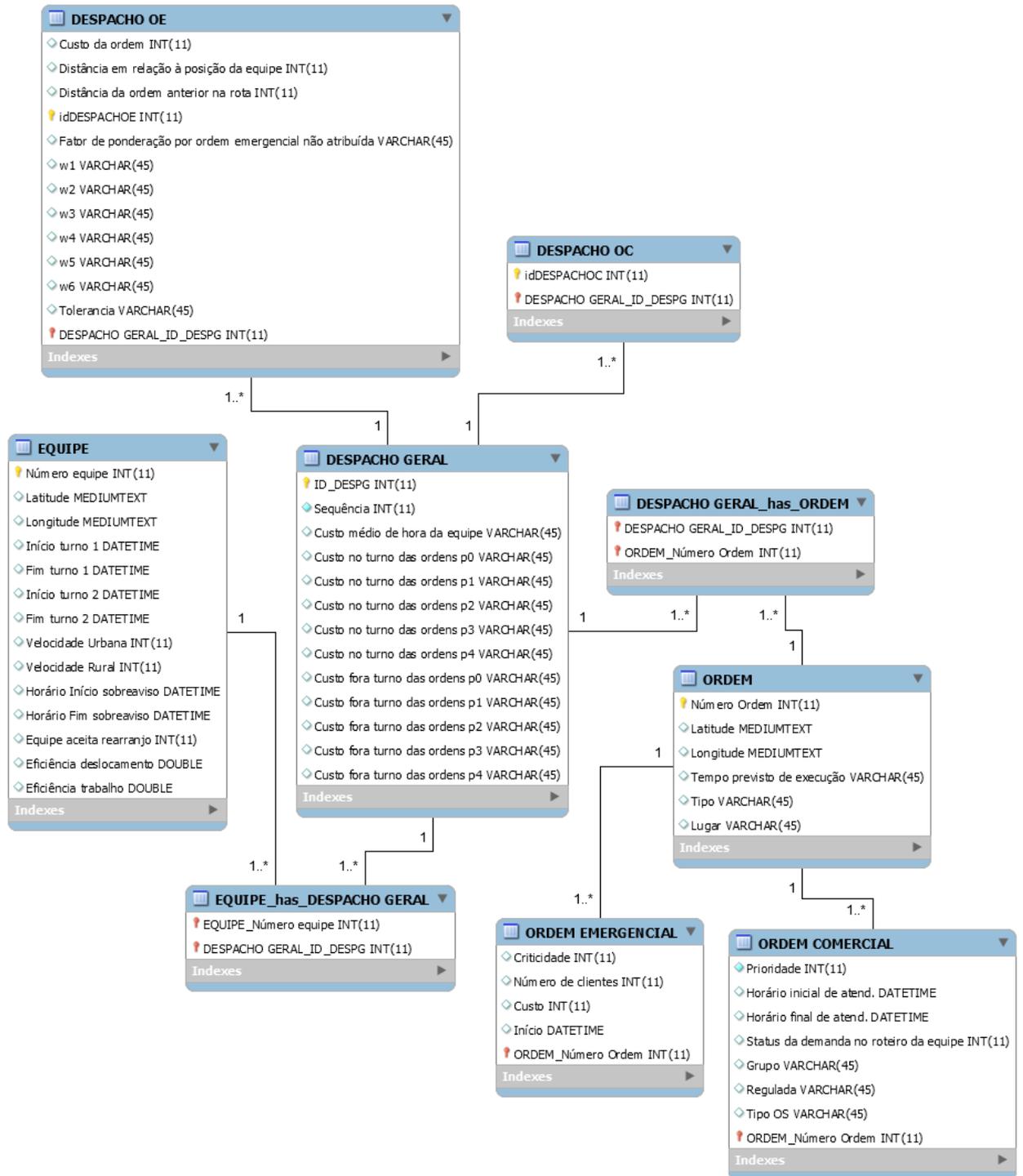
SHEN, M., CARSWELL, M., SANTHANAM, R., BAILEY, K. Emergency management information systems: Could decision makers be supported in choosing display formats?. **Decision Support Systems**, v. 52, n. 2, p. 318-330, 2012.

VAN DER LANS, R. F. **SQL for MySQL Developers: a comprehensive tutorial and reference**. Pearson Education. Boston. 2007.

# Apêndice 1 – Modelagem do ERD (notação UML)



## Apêndice 2 – ERD completo desenhado no Workbench



### Apêndice 3 – Script de comando SQL para criação das tabelas

```
-- MySQL Script generated by MySQL Workbench
-- 11/05/15 21:32:06
-- Model: New Model   Version: 1.0
-- MySQL Workbench Forward Engineering
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `Teste` DEFAULT CHARACTER SET utf8 ;
USE `Teste` ;

-----
-- Table `DESPACHO GERAL`
-----
CREATE TABLE IF NOT EXISTS `DESPACHO GERAL` (
  `ID_DESPG` INT(11) NOT NULL ",
  `Sequência` INT(11) NOT NULL ",
  `Custo médio de hora da equipe` VARCHAR(45) NULL DEFAULT NULL ", `Custo no
turno das ordens p0` VARCHAR(45) NULL DEFAULT NULL ", `Custo no turno das
ordens p1` VARCHAR(45) NULL DEFAULT NULL ", `Custo no turno das ordens p2`
VARCHAR(45) NULL DEFAULT NULL ", `Custo no turno das ordens p3`
VARCHAR(45) NULL DEFAULT NULL ", `Custo no turno das ordens p4`
VARCHAR(45) NULL DEFAULT NULL ", `Custo fora turno das ordens p0`
VARCHAR(45) NULL DEFAULT NULL ", `Custo fora turno das ordens p1`
VARCHAR(45) NULL DEFAULT NULL ", `Custo fora turno das ordens p2`
VARCHAR(45) NULL DEFAULT NULL ", `Custo fora turno das ordens p3`
VARCHAR(45) NULL DEFAULT NULL ", `Custo fora turno das ordens p4`
VARCHAR(45) NULL DEFAULT NULL ", PRIMARY KEY (`ID_DESPG`) ")
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

-----
-- Table `ORDEM`
-----
CREATE TABLE IF NOT EXISTS `ORDEM` ( `Número Ordem` INT(11) NOT NULL
AUTO_INCREMENT ", `Latitude` MEDIUMTEXT NULL DEFAULT NULL ",
`Longitude` MEDIUMTEXT NULL DEFAULT NULL ",
  `Tempo previsto de execução` VARCHAR(45) NULL DEFAULT NULL ", `Tipo`
VARCHAR(45) NULL DEFAULT NULL ", `Lugar` VARCHAR(45) NULL DEFAULT
NULL ", PRIMARY KEY (`Número Ordem`) ")
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

-----
-- Table `DESPACHO GERAL_has_ORDEM`
-----
CREATE TABLE IF NOT EXISTS `DESPACHO GERAL_has_ORDEM` ( `DESPACHO
GERAL_ID_DESPG` INT(11) NOT NULL ", `ORDEM_Número Ordem` INT(11) NOT
```

```

NULL ", PRIMARY KEY (`DESPACHO GERAL_ID_DESPG`, `ORDEM_Número
Ordem`) ", INDEX `fk_DESPACHO GERAL_has_ORDEM_ORDEM1_idx`
(`ORDEM_Número Ordem` ASC) ", INDEX `fk_DESPACHO
GERAL_has_ORDEM_DESPACHO GERAL1_idx` (`DESPACHO GERAL_ID_DESPG`
ASC) ", CONSTRAINT `fk_DESPACHO GERAL_has_ORDEM_DESPACHO GERAL1`
FOREIGN KEY (`DESPACHO GERAL_ID_DESPG`)
REFERENCES `DESPACHO GERAL` (`ID_DESPG`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_DESPACHO GERAL_has_ORDEM_ORDEM1`
FOREIGN KEY (`ORDEM_Número Ordem`)
REFERENCES `ORDEM` (`Número Ordem`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `DESPACHO OC`

```

```

-----
CREATE TABLE IF NOT EXISTS `DESPACHO OC` ( `idDESPACHOC` INT(11) NOT
NULL AUTO_INCREMENT ", `DESPACHO GERAL_ID_DESPG` INT(11) NOT NULL
", PRIMARY KEY (`idDESPACHOC`, `DESPACHO GERAL_ID_DESPG`) ",
INDEX `fk_DESPACHO OC_DESPACHO GERAL1_idx` (`DESPACHO
GERAL_ID_DESPG` ASC) ",
CONSTRAINT `fk_DESPACHO OC_DESPACHO GERAL1`
FOREIGN KEY (`DESPACHO GERAL_ID_DESPG`)
REFERENCES `DESPACHO GERAL` (`ID_DESPG`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `DESPACHO OE`

```

```

-----
CREATE TABLE IF NOT EXISTS `DESPACHO OE` ( `Custo da ordem` INT(11) NULL
DEFAULT NULL ", `Distância em relação à posição da equipe` INT(11) NULL DEFAULT
NULL ", `Distância da ordem anterior na rota` INT(11) NULL DEFAULT NULL ",
`idDESPACHOE` INT(11) NOT NULL AUTO_INCREMENT ", `Fator de ponderação por
ordem emergencial não atribuída` VARCHAR(45) NULL DEFAULT NULL ", `w1`
VARCHAR(45) NULL DEFAULT NULL ", `w2` VARCHAR(45) NULL DEFAULT
NULL ", `w3` VARCHAR(45) NULL DEFAULT NULL ", `w4` VARCHAR(45) NULL
DEFAULT NULL ",
`w5` VARCHAR(45) NULL DEFAULT NULL ", `w6` VARCHAR(45) NULL
DEFAULT NULL ", `Tolerancia` VARCHAR(45) NULL DEFAULT NULL ",
`DESPACHO GERAL_ID_DESPG` INT(11) NOT NULL ",
PRIMARY KEY (`idDESPACHOE`, `DESPACHO GERAL_ID_DESPG`) ", INDEX
`fk_DESPACHO OE_DESPACHO GERAL1_idx` (`DESPACHO GERAL_ID_DESPG`
ASC) ",
CONSTRAINT `fk_DESPACHO OE_DESPACHO GERAL1`
FOREIGN KEY (`DESPACHO GERAL_ID_DESPG`)

```

```

REFERENCES `DESPACHO GERAL` (`ID_DESPG`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `EQUIPE`
-----
CREATE TABLE IF NOT EXISTS `EQUIPE` ( `Número equipe` INT(11) NOT NULL
AUTO_INCREMENT ", `Latitude` MEDIUMTEXT NULL DEFAULT NULL ",
`Longitude` MEDIUMTEXT NULL DEFAULT NULL ",
`Início turno 1` DATETIME NULL DEFAULT NULL ", `Fim turno 1` DATETIME
NULL DEFAULT NULL ", `Início turno 2` DATETIME NULL DEFAULT NULL ", `Fim
turno 2` DATETIME NULL DEFAULT NULL ",
`Velocidade Urbana` INT(11) NULL DEFAULT NULL ", `Velocidade Rural` INT(11)
NULL DEFAULT NULL ", `Horário Início sobreaviso` DATETIME NULL DEFAULT
NULL ", `Horário Fim sobreaviso` DATETIME NULL DEFAULT NULL ", `Equipe aceita
rearranjo` INT(11) NULL DEFAULT NULL ",
`Eficiência deslocamento` DOUBLE NULL DEFAULT NULL ", `Eficiência trabalho`
DOUBLE NULL DEFAULT NULL ", PRIMARY KEY (`Número equipe`) ")
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `EQUIPE_has_DESPACHO GERAL`
-----
CREATE TABLE IF NOT EXISTS `EQUIPE_has_DESPACHO GERAL` (
`EQUIPE_Número equipe` INT(11) NOT NULL ", `DESPACHO GERAL_ID_DESPG`
INT(11) NOT NULL DEFAULT '0' ", PRIMARY KEY (`EQUIPE_Número equipe`,
`DESPACHO GERAL_ID_DESPG`) ",
INDEX `fk_EQUIPE_has_DESPACHO GERAL_DESPACHO GERAL1_idx`
(`DESPACHO GERAL_ID_DESPG` ASC) ",
INDEX `fk_EQUIPE_has_DESPACHO GERAL_EQUIPE_idx` (`EQUIPE_Número
equipe` ASC) ",
CONSTRAINT `fk_EQUIPE_has_DESPACHO GERAL_DESPACHO GERAL1`
FOREIGN KEY (`DESPACHO GERAL_ID_DESPG`)
REFERENCES `DESPACHO GERAL` (`ID_DESPG`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_EQUIPE_has_DESPACHO GERAL_EQUIPE`
FOREIGN KEY (`EQUIPE_Número equipe`)
REFERENCES `EQUIPE` (`Número equipe`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `ORDEM COMERCIAL`
-----
CREATE TABLE IF NOT EXISTS `ORDEM COMERCIAL` ( `Prioridade` INT(11) NOT
NULL ", `Horário inicial de atend.` DATETIME NULL DEFAULT NULL ",

```

```

`Horário final de atend.` DATETIME NULL DEFAULT NULL ",
`Status da demanda no roteiro da equipe` INT(11) NULL DEFAULT NULL ", `Grupo`
VARCHAR(45) NULL DEFAULT NULL ", `Regulada` VARCHAR(45) NULL DEFAULT
NULL ", `Tipo OS` VARCHAR(45) NULL DEFAULT NULL ", `ORDEM_Número
Ordem` INT(11) NOT NULL ", PRIMARY KEY (`ORDEM_Número Ordem` ) ",
CONSTRAINT `fk_ORDEM COMERCIAL_ORDEM1`
FOREIGN KEY (`ORDEM_Número Ordem`)
REFERENCES `ORDEM` (`Número Ordem`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
-----
-- Table `ORDEM EMERGENCIAL`
-----
CREATE TABLE IF NOT EXISTS `ORDEM EMERGENCIAL` ( `Criticidade` INT(11)
NULL DEFAULT NULL ", `Número de clientes` INT(11) NULL DEFAULT NULL ",
`Custo` INT(11) NULL DEFAULT NULL ", `Início` DATETIME NULL DEFAULT
NULL ", `ORDEM_Número Ordem` INT(11) NOT NULL ", PRIMARY KEY
(`ORDEM_Número Ordem` ) ",
CONSTRAINT `fk_ORDEM EMERGENCIAL_ORDEM1`
FOREIGN KEY (`ORDEM_Número Ordem`)
REFERENCES `ORDEM` (`Número Ordem`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

#### Apêndice 4 - Script de comando SQL para inserção dos dados

Inserção de equipes:

```
INSERT INTO `EQUIPE` (`Número equipe`, `Latitude`, `Longitude`, `Início turno 1` `Fim
turno 1`, `Início turno 2`, `Fim turno 2`, `Velocidade Urbana`, `Velocidade Rural`, `Horário
Início sobreaviso`, `Horário Fim sobreaviso`, `Equipe aceita rearranjo`, `Eficiência
deslocamento`, `Eficiência trabalho`)
VALUES
(237, '29890370000', '51135860000', '2015-03-17 18:00:00', '2015-03-17 22:00:00', '2015-03-
18 13:00:00', '2015-03-18 17:00:00', 0, 0, NULL, NULL, 0, 500, 500), (263, '29899710000',
'51140690000', '2015-03-17 17:00:00', '2015-03-17 21:00:00', '2015-03-18 12:00:00', '2015-
03-18 16:00:00', 0, 0, NULL, NULL, 0, 500, 500), (265, '29889090000', '51131540000',
'2015-03-17 17:00:00', '2015-03-17 22:00:00', '2015-03-18 13:00:00', '2015-03-18 16:00:00',
0, 0, NULL, NULL, 0, 500, 500);
```

Inserção de ordens:

```
INSERT INTO `ORDEM` (`Número Ordem`, `Latitude`, `Longitude`, `Tempo previsto de
execução`, `Tipo`, `Lugar`) VALUES
(4527835, '-2.985.271.500', '-5.117.626.300', NULL, 'E', NULL), (4527864, '-2.984.263.081',
'-5.114.549.751', NULL, 'E', NULL), (4527898, '-2.985.212.800', '-5.117.534.000', NULL, 'E',
NULL), (4527900, '-2.983.615.306', '-5.115.861.897', NULL, 'E', NULL), (4527932, '-
2.983.823.000', '-5.114.536.500', NULL, 'E', NULL), (4527986, '-2.983.750.700', '-
5.114.162.300', NULL, 'E', NULL), (4527989, '-2.988.510.065', '-5.118.877.279', NULL, 'E',
NULL), (4528006, '-2.984.883.083', '-5.118.077.622', NULL, 'E', NULL), (4528017, '-
2.984.892.500', '-5.118.568.800', NULL, 'E', NULL), (4528054, '-2.984.169.800', '-
5.117.327.800', NULL, 'E', NULL), (4528173, '-2.983.594.300', '-5.116.201.700', NULL, 'E',
NULL), (4528214, '-2.984.892.500', '-5.118.568.800', NULL, 'E', NULL), (4528382, '-
2.983.734.500', '-5.114.028.200', NULL, 'E', NULL), (4528445, '-2.985.232.500', '-
5.115.475.300', NULL, 'E', NULL), (4528696, '-2.984.832.608', '-5.119.028.840', NULL, 'E',
NULL), (4529065, '-2.984.547.800', '-5.114.411.800', NULL, 'E', NULL), (4529071, '-
2.984.711.700', '-5.114.393.000', NULL, 'E', NULL), (4529177, '-2.984.070.632', '-
5.114.152.651', NULL, 'E', NULL), (4529221, '-2.984.369.300', '-5.114.301.800', NULL, 'E',
NULL), (4529439, '-2.983.275.300', '-5.115.779.800', NULL, 'E', NULL), (4529456, '-
2.983.759.500', '-5.114.703.700', NULL, 'E', NULL), (4529561, '-2.982.920.500', '-
5.114.527.200', NULL, 'E', NULL), (4529567, '-2.984.041.200', '-5.114.046.300', NULL, 'E',
NULL), (4529577, '-2.982.920.500', '-5.114.527.200', NULL, 'E', NULL), (4529687, '-
2.982.557.540', '-5.116.598.775', NULL, 'E', NULL), (4529711, '-2.982.451.977', '-
5.116.561.523', NULL, 'E', NULL), (4529713, '-2.984.425.200', '-5.115.097.000', NULL, 'E',
NULL), (4530045, '-2.984.202.000', '-5.114.408.000', NULL, 'E', NULL), (4530172, '-
2.984.568.082', '-5.116.082.288', NULL, 'E', NULL), (4530228, '-2.985.232.500', '-
5.115.475.300', NULL, 'E', NULL), (4530496, '-2.984.686.700', '-5.115.105.700', NULL, 'E',
NULL), (4530516, '-2.984.938.452', '-5.116.986.645', NULL, 'E', NULL), (4530622, '-
2.984.652.800', '-5.114.190.300', NULL, 'E', NULL), (4530659, '-2.982.196.300', '-
5.115.670.800', NULL, 'E', NULL), (4530735, '-2.984.766.591', '-5.114.800.324', NULL, 'E',
NULL), (4530988, '-2.984.273.092', '-5.116.440.809', NULL, 'E', NULL), (4531047, '-
2.982.731.700', '-5.116.771.700', NULL, 'E', NULL), (4531126, '-2.984.336.300', '-
5.115.527.500', NULL, 'E', NULL), (4531269, '-2.984.318.500', '-5.115.945.800', NULL, 'E',
NULL), (4531302, '-2.984.833.200', '-5.113.826.000', NULL, 'E', NULL), (4531488, '-
```

2.988.930.023', '-5.118.174.281', NULL, 'E', NULL), (4531862, '-2.976.665.800', '-  
5.114.121.800', NULL, 'E', NULL), (4532371, '-2.988.671.190', '-5.118.721.337', NULL, 'E',  
NULL), (4532409, '-2.988.696.000', '-5.118.867.500', NULL, 'E', NULL), (4532417, '-  
2.984.686.700', '-5.115.105.700', NULL, 'E', NULL), (4533020, '-2.990.562.700', '-  
5.116.369.200', NULL, 'E', NULL), (4533067, '-2.984.266.000', '-5.114.227.200', NULL, 'E',  
NULL), (4533072, '-2.985.638.500', '-5.114.697.500', NULL, 'E', NULL), (4533174, '-  
2.984.686.700', '-5.115.105.700', NULL, 'E', NULL), (4533209, '-2.984.471.300', '-  
5.114.268.500', NULL, 'E', NULL), (4533229, '-2.982.196.300', '-5.115.670.800', NULL, 'E',  
NULL), (4533288, '-2.982.731.700', '-5.116.771.700', NULL, 'E', NULL), (4533597, '-  
2.983.863.500', '-5.116.954.500', NULL, 'E', NULL), (4533755, '-2.984.097.448', '-  
5.116.397.741', NULL, 'E', NULL), (4533868, '-2.983.353.618', '-5.115.170.543', NULL, 'E',  
NULL), (4534109, '-2.984.097.448', '-5.116.397.741', NULL, 'E', NULL), (4534197, '-  
2.985.851.200', '-5.114.662.500', NULL, 'E', NULL), (4534544, '-2.988.519.179', '-  
5.118.712.472', NULL, 'E', NULL), (4534551, '-2.986.058.021', '-5.114.937.134', NULL, 'E',  
NULL), (4534759, '-2.985.513.500', '-5.115.377.300', NULL, 'E', NULL), (4534885, '-  
2.986.251.000', '-5.113.266.000', NULL, 'E', NULL), (4534887, '-2.986.049.300', '-  
5.114.830.800', NULL, 'E', NULL), (4534905, '-2.988.647.700', '-5.118.665.800', NULL, 'E',  
NULL), (4534930, '-2.988.693.800', '-5.118.891.500', NULL, 'E', NULL), (4534974, '-  
2.988.671.190', '-5.118.721.337', NULL, 'E', NULL), (4535018, '-2.988.770.500', '-  
5.118.860.200', NULL, 'E', NULL), (4535039, '-2.988.807.386', '-5.118.732.822', NULL, 'E',  
NULL), (4535228, '-2.986.311.642', '-5.114.856.073', NULL, 'E', NULL), (4535287, '-  
2.983.345.500', '-5.115.525.700', NULL, 'E', NULL), (4535410, '-2.984.264.800', '-  
5.114.226.300', NULL, 'E', NULL), (4535447, '-2.985.510.500', '-5.112.995.000', NULL, 'E',  
NULL), (4535501, '-2.986.251.000', '-5.113.266.000', NULL, 'E', NULL), (4535520, '-  
2.984.292.500', '-5.116.532.700', NULL, 'E', NULL), (4535674, '-2.983.461.300', '-  
5.116.695.800', NULL, 'E', NULL), (4535731, '-2.986.049.300', '-5.114.830.800', NULL, 'E',  
NULL), (4536115, '-2.984.855.519', '-5.114.148.930', NULL, 'E', NULL), (4536292, '-  
2.982.512.000', '-5.115.163.262', NULL, 'E', NULL), (4536307, '-2.984.293.200', '-  
5.116.532.800', NULL, 'E', NULL), (4536363, '-2.984.264.800', '-5.114.226.300', NULL, 'E',  
NULL), (4536463, '-2.984.813.000', '-5.114.182.800', NULL, 'E', NULL), (4536501, '-  
2.984.652.800', '-5.114.190.300', NULL, 'E', NULL), (4536556, '-2.986.244.000', '-  
5.115.005.500', NULL, 'E', NULL), (4536659, '-2.982.206.000', '-5.115.829.200', NULL, 'E',  
NULL), (4536756, '-2.986.301.700', '-5.114.923.500', NULL, 'E', NULL), (4536785, '-  
2.986.117.500', '-5.114.972.200', NULL, 'E', NULL), (4537046, '-2.986.301.700', '-  
5.114.923.500', NULL, 'E', NULL), (4537152, '-2.983.505.700', '-5.115.245.200', NULL, 'E',  
NULL), (4537196, '-2.984.292.500', '-5.116.532.700', NULL, 'E', NULL), (85024033, '-  
5.122.590.000', '-2.991.089.000', 30, 'C', NULL), (85024077, '-5.122.408.000', '-  
2.979.489.000', 30, 'C', NULL), (85024082, '-5.115.326.000', '-2.985.253.000', 30, 'C',  
NULL), (85024087, '-5.114.430.000', '-2.991.854.000', 30, 'C', NULL), (85024093, '-  
5.114.357.000', '-2.990.643.000', 30, 'C', NULL), (85024094, '-5.115.114.000', '-  
2.984.844.000', 30, 'C', NULL), (85024095, '-5.114.898.000', '-2.992.138.000', 30, 'C',  
NULL), (85024099, '-5.115.064.000', '-2.985.131.000', 30, 'C', NULL), (85024118, '-  
5.113.095.000', '-2.988.757.000', 30, 'C', NULL), (85024126, '-5.117.158.000', '-  
2.990.066.000', 30, 'C', NULL), (85024241, '-5.129.057.000', '-2.987.315.000', 30, 'C',  
NULL), (85024243, '-5.128.258.000', '-2.987.479.000', 30, 'C', NULL), (85024261, '-  
5.118.607.000', '-2.992.051.000', 30, 'C', NULL), (85024264, '-5.118.674.000', '-  
2.991.964.000', 30, 'C', NULL), (85024265, '-5.120.664.000', '-2.991.514.000', 30, 'C',  
NULL), (85024315, '-5.115.882.000', '-2.985.538.000', 30, 'C', NULL), (85024327, '-  
5.115.874.000', '-2.985.638.000', 30, 'C', NULL), (85024345, '-5.117.690.000', '-  
2.991.890.000', 30, 'C', NULL), (85024348, '-5.117.741.000', '-2.991.660.000', 30, 'C',

NULL), (85024385, '-5.125.193.000', '-2.988.401.000', 30, 'C', NULL), (85024415, '-5.125.971.000', '-2.986.360.000', 30, 'C', NULL), (85024432, '-5.117.530.000', '-2.984.772.000', 30, 'C', NULL), (85024433, '-5.120.842.000', '-2.990.506.000', 30, 'C', NULL), (85024443, '-5.120.422.000', '-2.989.808.000', 30, 'C', NULL), (85024448, '-5.120.307.000', '-2.989.900.000', 30, 'C', NULL), (85024476, '-5.119.153.000', '-2.995.636.000', 30, 'C', NULL), (85024477, '-5.117.750.000', '-2.994.814.000', 30, 'C', NULL), (85024478, '-5.118.142.000', '-2.994.843.000', 30, 'C', NULL), (85024479, '-5.118.818.000', '-2.994.490.000', 30, 'C', NULL), (85024482, '-5.119.149.000', '-2.995.659.000', 30, 'C', NULL), (85024488, '-5.118.828.000', '-2.994.773.000', 30, 'C', NULL), (85024491, '-5.117.927.000', '-2.991.912.000', 30, 'C', NULL), (85024493, '-5.118.151.000', '-2.991.905.000', 30, 'C', NULL), (85024495, '-5.118.517.000', '-2.991.182.000', 30, 'C', NULL), (85024497, '-5.118.239.000', '-2.991.536.000', 30, 'C', NULL), (85024502, '-5.117.971.000', '-2.991.480.000', 30, 'C', NULL), (85024507, '-5.118.867.000', '-2.991.282.000', 30, 'C', NULL), (85024542, '-5.116.801.000', '-2.995.257.000', 30, 'C', NULL), (85024544, '-5.116.026.000', '-2.995.043.000', 30, 'C', NULL), (85024546, '-5.117.115.000', '-2.985.847.000', 30, 'C', NULL), (85024577, '-5.113.318.000', '-2.989.386.000', 30, 'C', NULL), (85024713, '-5.119.160.000', '-2.995.561.000', 30, 'C', NULL), (85024758, '-5.119.164.000', '-2.993.275.000', 30, 'C', NULL), (85024761, '-5.118.473.000', '-2.991.875.000', 30, 'C', NULL), (85024762, '-5.118.564.000', '-2.992.054.000', 30, 'C', NULL), (85024767, '-5.121.891.000', '-2.991.343.000', 30, 'C', NULL), (85024768, '-5.120.561.000', '-2.991.352.000', 30, 'C', NULL), (85024785, '-5.122.941.000', '-2.990.877.000', 30, 'C', NULL), (85024832, '-5.129.774.000', '-2.984.078.000', 30, 'C', NULL), (85024869, '-5.116.527.000', '-2.993.711.000', 30, 'C', NULL), (85024915, '-5.113.579.000', '-2.989.223.000', 30, 'C', NULL), (85024919, '-5.114.156.000', '-2.989.152.000', 30, 'C', NULL), (85024955, '-5.116.966.000', '-2.988.108.000', 30, 'C', NULL), (85025024, '-5.116.680.000', '-2.984.904.000', 30, 'C', NULL), (85025038, '-5.119.414.000', '-2.996.193.000', 30, 'C', NULL), (85025039, '-5.120.056.000', '-2.996.282.000', 30, 'C', NULL), (85025067, '-5.119.617.000', '-2.992.032.000', 30, 'C', NULL), (85025070, '-5.119.753.000', '-2.992.456.000', 30, 'C', NULL), (85025071, '-5.119.505.832', '-2.993.154.955', 30, 'C', NULL), (85025072, '-5.119.921.000', '-2.992.152.000', 30, 'C', NULL), (85025083, '-5.121.686.000', '-2.990.826.000', 30, 'C', NULL), (85025092, '-5.122.289.000', '-2.991.020.000', 30, 'C', NULL), (85025095, '-5.119.470.000', '-2.991.314.000', 30, 'C', NULL), (85025123, '-5.115.478.000', '-2.996.055.000', 30, 'C', NULL), (85025128, '-5.116.923.000', '-2.995.197.000', 30, 'C', NULL), (85025129, '-5.116.661.000', '-2.995.138.000', 30, 'C', NULL), (85025136, '-5.116.650.000', '-2.995.084.000', 30, 'C', NULL), (85025173, '-5.116.405.000', '-2.992.560.000', 30, 'C', NULL), (85025210, '-5.114.036.000', '-2.989.278.000', 30, 'C', NULL), (85025219, '-5.113.939.000', '-2.989.139.000', 30, 'C', NULL), (85025227, '-5.113.016.000', '-2.989.165.000', 30, 'C', NULL), (85025229, '-5.113.054.000', '-2.989.053.000', 30, 'C', NULL), (85025231, '-5.113.064.000', '-2.989.016.000', 30, 'C', NULL), (85025237, '-5.113.083.000', '-2.988.812.000', 30, 'C', NULL), (85025238, '-5.113.120.000', '-2.988.908.000', 30, 'C', NULL), (85025241, '-5.112.975.232', '-2.989.390.564', 30, 'C', NULL), (85025242, '-5.113.408.000', '-2.989.582.000', 30, 'C', NULL), (85025264, '-5.125.211.000', '-2.987.328.000', 30, 'C', NULL), (85025265, '-5.125.384.000', '-2.987.662.000', 30, 'C', NULL), (85025283, '-5.117.522.000', '-2.984.039.000', 30, 'C', NULL), (85025353, '-5.121.555.000', '-2.991.402.000', 30, 'C', NULL), (85025442, '-5.115.345.000', '-2.986.088.000', 30, 'C', NULL), (85025495, '-5.125.585.000', '-2.988.491.000', 30, 'C', NULL), (85025507, '-5.116.507.000', '-2.988.004.000', 30, 'C', NULL), (85025512, '-5.120.033.000', '-2.990.688.000', 30, 'C', NULL), (85025513, '-5.119.485.000', '-

2.990.284.000', 30, 'C', NULL), (85025516, '-5.118.106.000', '-2.990.339.000', 30, 'C', NULL), (85025520, '-5.116.367.000', '-2.990.563.000', 30, 'C', NULL), (85025521, '-5.119.954.000', '-2.990.693.000', 30, 'C', NULL), (85025522, '-5.122.551.000', '-2.990.462.000', 30, 'C', NULL);

Inserção de ordens emergenciais:

```
INSERT INTO `ORDEM EMERGENCIAL`(`Criticidade`, `Número de clientes`, `Custo`, `Início`, `ORDEM_Número Ordem`) VALUES
(NULL, 1, NULL, '22/12/2013 10:13', 4527672), (NULL, 1, NULL, '22/12/2013 12:13', 4527818), (NULL, 1, NULL, '22/12/2013 16:57', 4527835), (NULL, 1156, NULL, '22/12/2013 17:51', 4527864), (NULL, 1, NULL, '22/12/2013 18:24', 4527898), (NULL, 3720, NULL, '22/12/2013 19:28', 4527900), (NULL, 1, NULL, '22/12/2013 19:37', 4527932), (NULL, 1, NULL, '22/12/2013 21:52', 4527986), (NULL, 1, NULL, '22/12/2013 21:02', 4527989), (NULL, 60, NULL, '23/12/2013 00:03', 4528006), (NULL, 1, NULL, '22/12/2013 19:50', 4528017), (NULL, 1, NULL, '22/12/2013 19:42', 4528054), (NULL, 1, NULL, '23/12/2013 09:14', 4528173), (NULL, 1, NULL, '23/12/2013 09:09', 4528214), (NULL, 1, NULL, '23/12/2013 14:13', 4528382), (NULL, 1, NULL, '23/12/2013 15:05', 4528445), (NULL, 1, NULL, '23/12/2013 20:04', 4528696), (NULL, 1, NULL, '24/12/2013 14:46', 4529065), (NULL, 1, NULL, '24/12/2013 14:56', 4529071), (NULL, 1, NULL, '24/12/2013 18:23', 4529177), (NULL, 1, NULL, '24/12/2013 19:35', 4529221), (NULL, 1, NULL, '25/12/2013 13:08', 4529439), (NULL, 1, NULL, '25/12/2013 13:31', 4529456), (NULL, 1, NULL, '25/12/2013 18:18', 4529561), (NULL, 1, NULL, '25/12/2013 17:40', 4529567), (NULL, 1, NULL, '25/12/2013 19:27', 4529577), (NULL, 95, NULL, '26/12/2013 00:59', 4529687), (NULL, 70, NULL, '26/12/2013 06:14', 4529711), (NULL, 1, NULL, '25/12/2013 21:17', 4529713), (NULL, 1, NULL, '26/12/2013 09:10', 4530045), (NULL, 1, NULL, '26/12/2013 14:27', 4530172), (NULL, 1, NULL, '26/12/2013 13:37', 4530228), (NULL, 1, NULL, '26/12/2013 17:16', 4530496), (NULL, 1, NULL, '26/12/2013 16:56', 4530516), (NULL, 1, NULL, '26/12/2013 19:08', 4530622), (NULL, 1, NULL, '26/12/2013 17:46', 4530659), (NULL, 73, NULL, '26/12/2013 20:35', 4530735), (NULL, 16, NULL, '27/12/2013 08:01', 4530988), (NULL, 1, NULL, '26/12/2013 21:13', 4531047), (NULL, 1, NULL, '26/12/2013 23:51', 4531126), (NULL, 1, NULL, '27/12/2013 07:08', 4531269), (NULL, 1, NULL, '27/12/2013 10:42', 4531302), (NULL, 1, NULL, '27/12/2013 13:37', 4531488), (NULL, 1, NULL, '27/12/2013 14:02', 4531862), (NULL, 100, NULL, '27/12/2013 21:38', 4532371), (NULL, 1, NULL, '27/12/2013 21:18', 4532409), (NULL, 1, NULL, '27/12/2013 16:59', 4532417), (NULL, 1, NULL, '27/12/2013 20:14', 4533020), (NULL, 1, NULL, '27/12/2013 21:14', 4533067), (NULL, 1, NULL, '28/12/2013 09:48', 4533072), (NULL, 1, NULL, '28/12/2013 14:00', 4533174), (NULL, 1, NULL, '28/12/2013 13:59', 4533209), (NULL, 1, NULL, '28/12/2013 11:25', 4533229), (NULL, 1, NULL, '28/12/2013 12:01', 4533288), (NULL, 1, NULL, '28/12/2013 17:35', 4533597), (NULL, 85, NULL, '28/12/2013 21:45', 4533755), (NULL, 175, NULL, '29/12/2013 06:40', 4533868), (NULL, 85, NULL, '29/12/2013 15:24', 4534109), (NULL, 1, NULL, '29/12/2013 16:17', 4534197), (NULL, 476, NULL, '30/12/2013 02:30', 4534544), (NULL, 307, NULL, '30/12/2013 02:40', 4534551), (NULL, 1, NULL, '30/12/2013 08:25', 4534759), (NULL, 1, NULL, '30/12/2013 08:26', 4534885), (NULL, 1, NULL, '30/12/2013 08:06', 4534887), (NULL, 1, NULL, '30/12/2013 09:36', 4534905), (NULL, 1, NULL, '30/12/2013 08:37', 4534930), (NULL, 100, NULL, '30/12/2013 10:01', 4534974), (NULL, 1, NULL, '30/12/2013 10:46', 4535018), (NULL, 53, NULL, '30/12/2013 10:49', 4535039), (NULL, 180, NULL, '30/12/2013 02:40', 4535228), (NULL, 1, NULL, '30/12/2013 13:34', 4535287), (NULL, 1, NULL, '30/12/2013 11:39', 4535410), (NULL, 1, NULL, '30/12/2013 13:16', 4535447), (NULL, 1, NULL, '30/12/2013
```

12:15', 4535501), (NULL, 1, NULL, '30/12/2013 12:54', 4535520), (NULL, 1, NULL, '30/12/2013 15:50', 4535674), (NULL, 1, NULL, '30/12/2013 12:02', 4535731), (NULL, 144, NULL, '30/12/2013 22:33', 4536115), (NULL, 1, NULL, '30/12/2013 15:46', 4536292), (NULL, 1, NULL, '30/12/2013 18:47', 4536307), (NULL, 1, NULL, '30/12/2013 16:53', 4536363), (NULL, 1, NULL, '31/12/2013 08:00', 4536463), (NULL, 1, NULL, '30/12/2013 18:54', 4536501), (NULL, 1, NULL, '30/12/2013 22:13', 4536556), (NULL, 1, NULL, '31/12/2013 12:13', 4536659), (NULL, 1, NULL, '31/12/2013 12:17', 4536756), (NULL, 1, NULL, '31/12/2013 11:28', 4536785), (NULL, 1, NULL, '31/12/2013 16:04', 4537046), (NULL, 1, NULL, '31/12/2013 16:56', 4537152), (NULL, 1, NULL, '31/12/2013 17:48', 4537196);

Inserção de ordens comerciais:

```
INSERT INTO `ORDEM COMERCIAL` (`Prioridade`, `Horário inicial de atend.`, `Horário final de atend.`, `Status da demanda no roteiro da equipe`, `Grupo`, `Regulada`, `Tipo OS`, `ORDEM_Número Ordem`) VALUES
(NULL, '12/06/2013 00:00', '12/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024033),
(NULL, '19/06/2013 00:00', '19/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024077),
(NULL, '19/06/2013 00:00', '19/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024082),
(NULL, '19/06/2013 00:00', '05/04/2014 23:59', NULL, 'AES', 'N', 'TO508', 85024087),
(NULL, '19/06/2013 00:00', '07/01/2016 23:59', NULL, 'AES', 'N', 'TO508', 85024093),
(NULL, '19/06/2013 00:00', '19/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024094),
(NULL, '19/06/2013 00:00', '19/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024095),
(NULL, '19/06/2013 00:00', '19/07/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024099),
(NULL, '21/06/2013 00:00', '05/04/2014 23:59', NULL, 'AES', 'N', 'TO508', 85024118),
(NULL, '24/06/2013 00:00', '21/03/2014 23:59', NULL, 'AES', 'N', 'TO508', 85024126),
(NULL, '08/07/2013 00:00', '07/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024241),
(NULL, '08/07/2013 00:00', '07/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024243),
(NULL, '10/07/2013 00:00', '09/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024261),
(NULL, '10/07/2013 00:00', '09/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024264),
(NULL, '10/07/2013 00:00', '09/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024265),
(NULL, '16/07/2013 00:00', '15/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024315),
(NULL, '16/07/2013 00:00', '15/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024327),
(NULL, '18/07/2013 00:00', '17/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024345),
(NULL, '18/07/2013 00:00', '17/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024348),
(NULL, '23/07/2013 00:00', '22/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024385),
(NULL, '25/07/2013 00:00', '24/08/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024415),
(NULL, '02/08/2013 00:00', '01/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024432),
(NULL, '02/08/2013 00:00', '01/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024433),
(NULL, '02/08/2013 00:00', '01/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024443),
(NULL, '02/08/2013 00:00', '01/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024448),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024476),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024477),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024478),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024479),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024482),
(NULL, '07/08/2013 00:00', '06/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024488),
(NULL, '08/08/2013 00:00', '07/07/2015 23:59', NULL, 'AES', 'N', 'TO508', 85024491),
(NULL, '08/08/2013 00:00', '07/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024493),
(NULL, '08/08/2013 00:00', '07/09/2013 23:59', NULL, 'AES', 'N', 'TO508', 85024495),
```



```
(NULL, '26/11/2013 00:00', '26/12/2013 23:59', NULL, 'AES', 'N', 'TO508', 85025512),
(NULL, '26/11/2013 00:00', '27/03/2014 23:59', NULL, 'AES', 'N', 'TO508', 85025513),
(NULL, '26/11/2013 00:00', '27/03/2014 23:59', NULL, 'AES', 'N', 'TO508', 85025516),
(NULL, '26/11/2013 00:00', '26/12/2013 23:59', NULL, 'AES', 'N', 'TO508', 85025520),
(NULL, '26/11/2013 00:00', '26/12/2013 23:59', NULL, 'AES', 'N', 'TO508', 85025521),
(NULL, '26/11/2013 00:00', '26/12/2013 23:59', NULL, 'AES', 'N', 'TO508', 85025522);
```

Inserção dos despachos gerais:

```
INSERT INTO `DESPACHO GERAL` (`ID_DESPG`, `Sequência`, `Custo médio de hora da
equipe`, `Custo no turno das ordens p0`, `Custo no turno das ordens p1`, `Custo no turno das
ordens p2`, `Custo no turno das ordens p3`, `Custo no turno das ordens p4`, `Custo fora turno
das ordens p0`, `Custo fora turno das ordens p1`, `Custo fora turno das ordens p2`, `Custo
fora turno das ordens p3`, `Custo fora turno das ordens p4`)
VALUES ('1', '1', NULL, NULL),
(2, '2', NULL, NULL),
(3, '3', NULL, NULL),
(4, '4', NULL, NULL),
(5, '5', NULL, NULL);
```

Inserção dos despachos comerciais através dos despachos gerais:

```
INSERT INTO `DESPACHO OC` (`idDESPACHOC`, `DESPACHO GERAL_ID_DESPG`)
VALUES ('100', '4'), ('101', '5'), ('102', '3');
```

Inserção dos despachos emergenciais através dos despachos gerais:

```
INSERT INTO `DESPACHO OE` (`Custo da ordem`, `Distância em relação à posição da
equipe`, `Distância da ordem anterior na rota`, `idDESPACHOE`, `Fator de ponderação por
ordem emergencial não atribuída`, `w1`, `w2`, `w3`, `w4`, `w5`, `w6`, `Tolerancia`,
`DESPACHO GERAL_ID_DESPG`) VALUES (NULL, NULL, NULL, '200', NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, '1'), (NULL, NULL, NULL, '201',
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, '2');
```

Inserção dos registros da tabela Equipe\_DespachoGeral:

```
INSERT INTO `EQUIPE_has_DESPACHO GERAL` (`EQUIPE_Número equipe`,
`DESPACHO GERAL_ID_DESPG`) VALUES ('237', '1'), ('237', '4'), ('263', '2'), ('263', '5'),
('265', '3');
```

Inserção dos registros da tabela Despacho\_Ordem:

```
INSERT INTO `DESPACHO GERAL_has_ORDEM` (`DESPACHO GERAL_ID_DESPG`,
`ORDEM_Número Ordem`) VALUES ('1', '4527672'), ('1', '4527818'), ('1', '4527835'), ('1',
'4527864'), ('1', '4527898'), ('2', '4527900'), ('2', '4527932'), ('2', '4527986'), ('2', '4527989'),
('2', '4528006'), ('4', '85024033'), ('4', '85024077'), ('4', '85024082'), ('4', '85024087'), ('4',
'85024093'), ('4', '85024094'), ('4', '85024095'), ('4', '85024099'), ('4', '85024118'), ('4',
'85024126'), ('5', '85024241'), ('5', '85024243'), ('5', '85024261'), ('5', '85024264'), ('5',
'85024265'), ('5', '85024315'), ('5', '85024327'), ('5', '85024345'), ('5', '85024348'), ('5',
```

'85024385'), ('3', '85024415'), ('3', '85024432'), ('3', '85024433'), ('3', '85024443'), ('3', '85024448'), ('3', '85024476'), ('3', '85024477'), ('3', '85024478'), ('3', '85024479'), ('3', '85024482');

## Apêndice 5 - Script de comando SQL para criação das consultas

Linhas para criação do calendário:

```
select `DESPACHO GERAL_has_ORDEM`.`ORDEM_Número Ordem`,  
`EQUIPE_has_DESPACHO GERAL`.`EQUIPE_Número equipe`, `DESPACHO  
GERAL`.`ID_DESPG` from `DESPACHO GERAL_has_ORDEM` inner join `DESPACHO  
GERAL` on `DESPACHO GERAL`.`ID_DESPG` = `DESPACHO  
GERAL_has_ORDEM`.`DESPACHO GERAL_ID_DESPG` inner join  
`EQUIPE_has_DESPACHO GERAL` on `DESPACHO GERAL`.`ID_DESPG` =  
`EQUIPE_has_DESPACHO GERAL`.`DESPACHO GERAL_ID_DESPG`
```

Linhas de consulta para o somatório do Tempo previsto de execução:

```
select `EQUIPE_has_DESPACHO GERAL`.`EQUIPE_Número equipe` as `ID Equipe`,  
sum(`ORDEM`.`Tempo previsto de execução`) as `Tempo previsto de execução` from  
`DESPACHO GERAL_has_ORDEM` inner join `DESPACHO GERAL` on `DESPACHO  
GERAL`.`ID_DESPG` = `DESPACHO GERAL_has_ORDEM`.`DESPACHO  
GERAL_ID_DESPG` inner join `EQUIPE_has_DESPACHO GERAL` on `DESPACHO  
GERAL`.`ID_DESPG` = `EQUIPE_has_DESPACHO GERAL`.`DESPACHO  
GERAL_ID_DESPG` inner join `ORDEM` on `ORDEM`.`Número Ordem` = `DESPACHO  
GERAL_has_ORDEM`.`ORDEM_Número Ordem` group by `EQUIPE_has_DESPACHO  
GERAL`.`EQUIPE_Número equipe`
```